VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Sdílení souborů s využitím Zeroconf a protokolu Samba

File Sharing with Zeroconf and Protocol Samba

2020 Lukáš Majoros

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Zadání bakalářské práce

Student: **Lukáš Majoros**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2601R013 Telekomunikační technika

Téma: Sdílení souborů s využitím Zeroconf a protokolu Samba
File Sharing with Zeroconf and Protocol Samba

Jazyk vypracování: čeština

Zásady pro vypracování:

Cílem bakalářské práce je navrhnout a analyzovat sdílení souborů pomocí projektu Avahi a Samba v prostředí IPv6 protokolu. Řešení bude testováno ve virtualizovaném a embedded prostředí.

Řešení práce spočívá ve splnení následujících úkolů:
1. Popis protokolu Zeroconf.
2. Popis protikolů SMB a CIFS.
3. Návrh řešení s využitím virtualizovaného prostředí a embedded prostředí Raspberry Pi.
4. Ověření funkčnosti a analýza navrženého řešení.

Seznam doporučené odborné literatury:

[1] STEINBERG, Daniel, CHESHIRE, Stuart, *Zero Configuration Networking: The Definitive Guide.* O'Reilly Media 2005, ISBN-13: 978-0596101008
[2] GRAZIANI, Rick, *IPv6 Fundamentals: A Straightforward Approach to Understanding IPv6*Cisco Press 2017, ISBN-13: 978-1587144776

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Pavel Nevlud**

Datum zadání: 01.09.2019
Datum odevzdání: 30.04.2020

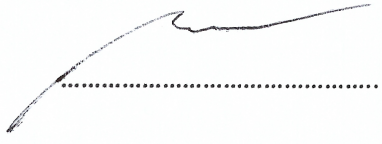
VYSOKÁ ŠKOLA BÁŇSKÁ-TECHNICKÁ UNIVERZITA Fakulta elektrotechniky a informatiky 1D16 OSTRAVA

prof. Ing. Miroslav Vozňák, Ph.D.
vedoucí katedry

prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

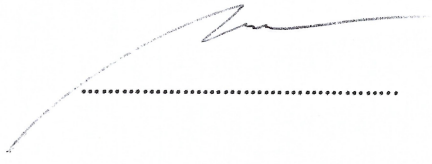
Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 30. dubna 2020



Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava.

V Ostravě 30. dubna 2020



Rád bych poděkoval vedoucímu této bakalářské práce Ing. Pavlu Nevludovi za užitečné rady, doporučení, poskytnutá zařízení a pravidelné konzultace. Své sestře PhDr. Nele Chobolové a přítelkyni Barboře Šamonilové za dodatečnou korekturu textu, ale hlavně své rodině za trpělivost.

Také nesmím opomenout poděkovat firmě SEIBERTMEDIA GmbH za bezplatné poskytnutí aplikace Draw.io, ve které byla zpracována většina obrázků z této bakalářské práce.

Abstrakt

Cílem bakalářské práce je navrhnout a analyzovat sdílení souborů pomocí projektu Avahi a Samba v prostředí IPv6 protokolu. Řešení bude testováno ve virtualizovaném a embedded prostředí. Teoretická část práce se bude věnovat popisu protokolu ZeroConf, popisu protokolů Samba a CIFS a návrhu řešení s využití virtualizovaného prostředí a embedded prostředí Raspberry PI.

Klíčová slova: Avahi; Bonjour; Zeroconf; mDNS; DNS-SD; IPv6; SMB; Samba; CIFS; Raspberry PI; Linux;

Abstract

The aim of this bachelor thesis is to design and analyze file sharing using the project Avahi and Samba in the environment of IPv6 protocol. The solution will be tested in a virtualized and embedded environment. The theoretical part of the thesis will be devoted to the description of the ZeroConf protocol, the description of the Samba and CIFS protocols and the design of the solution using the virtualized environment and embedded environment Raspberry PI.

Keywords: Avahi; Bonjour; Zeroconf; mDNS; DNS-SD; IPv6; SMB; Samba; CIFS; Raspberry PI; Linux;

Obsah

Seznam použitých zkratek a symbolů

AD	–	Active Directory
AES	–	Advanced Encryption Standard
AFRINIC	–	African Network Information Centre
ANY	–	Jakákoliv adresa / jméno
API	–	Application programming interface
APIPA	–	Automatic Private IP Addressing
APNIC	–	Asia Pacific Network Information Centre
ARIN	–	American Registry for Internet Numbers
CIDR	–	Classless Inter-Domain Routing
CIFS	–	Common Internet File System
CLI	–	Příkazový řádek
CNAME	–	Common name
CORBA	–	Common Object Request Broker Architecture
DC	–	Řadič domény
DHCP	–	Dynamic Host Configuration Protocol
DLNA	–	Digital Living Network Alliance
DNS	–	Domain Name System
DNS-SD	–	Domain Name System, Service - Discovery
DNSSEC	–	Domain Name System Security Extension
EIGRP	–	Enhanced Interior Gateway Routing Protocol
FEI	–	Fakulta elektrotechniky a informatik
FQDN	–	Fully qualified domain name
FTP	–	File Transer Protocol
GUI	–	Grafické rozhraní
HMACSHA-256	–	Keyed-hash Message Authentication Code Secure Hash Algorithm
HTML	–	Hypertext Markup Language
HTTP	–	Hypertext Transfer Protocol
HTTPS	–	Hypertext Transfer Protocol Secure
IAB	–	Internet Architecture Board
IANA	–	Internet Assigned Numbers Authority
ICMP	–	Internet Control Message Protocol
IEEE	–	Institute of Electrical and Electronics Engineers
IETF	–	Internet Engineering Task Force
IP	–	Internet Protocol
IP	–	Internet Protocol
IPP	–	Internet Printing Protocol

IPPs	–	Internet Printing Protocol
IPsec	–	Internet Protocol Security
IPv4	–	Internet Protocol version 4
IPv4 LL	–	Internet Protocol version 4 - Link Local
IPv6	–	Internet Protocol version 6
IPv6 LL	–	Internet Protocol version 6 - Link Local
IPX	–	Internetwork Packet Exchange
ISO/OSI	–	Referenční model ISO/OSI
IT	–	Informační technologie
JINI	–	Apache River
LACNIC	–	Latin American and Caribbean Internet Addresses Registry
LAN	–	Local Area Network
LGPL	–	Lesser General Public License
MD5	–	message-digest algorithm 5
mDNS	–	Multicast Domain Name System
MIIO	–	Xiaomi mDNS služba
MTU	–	Maximum transmission unit
NAT	–	Network Address Translation
NBF	–	NetBIOS Frames
NBT	–	NetBIOS over TCP/IP
NETBEUI	–	Network Basic Input/Output System
NETBIOS	–	Network Basic Input/Output System
NITS	–	Networking in the Small
NSEC	–	Next Secure record
OS	–	Operační Systém
OSPF	–	Open Shortest Path First
OSPFv3 DR	–	Open Shortest Path First verze 3 - Designated Router
OSPFv3	–	Open Shortest Path First verze 3
PC	–	Počítač
PDC	–	Primární řadič domény
PDL	–	Page description language
PIM	–	Protocol Independent Multicast
POSIX	–	Portable Operating System Interface
prefix	–	předpona, velikost sítě
PTR	–	Pointer
RFC	–	Request For Comments
RIPE NCC	–	Réseaux IP Européens Network Coordination Centre
RIPng	–	RIP další generace
RIR	–	regional Internet registry

RMI	–	Java remote method invocation
RPi	–	Raspberry PI
RTSP	–	Real Time Streaming Protocol
SHA	–	Secure Hash Algorithm
SLAAC	–	Stateless Address Autoconfiguration
SMB	–	Samba / Server Message Block
SOAP	–	Simple Object Access Protocol
SPX	–	Sequenced Packet Exchange
SPYC	–	Speaking out Your Certificate
SQL	–	Structured Query Language
SSH	–	Secure Shell
TCP	–	Transmission Control Protocol
TV	–	Televize
UDP	–	User Datagram Protocol
UPnP	–	Universal Plug and Play
URI	–	Uniform Resource Identifier
WAN	–	Wide Area Network
XML	–	Extensible Markup Language
Zeroconf	–	Zero Configuration

Seznam obrázků

Seznam tabulek

Seznam výpisů zdrojového kódu

Úvod

S neustále rostoucím počtem zařízení využívající TCP/IP sítě, je potřeba začít využívat nové typy technologií, pro jednodušší uživatelský přístup a kontrolu těchto zařízení. Tato bakalářská práce je zaměřena na jednu z těchto technologií pro jednodušší správu síťových zařízení a sdílení služeb po síti. Jedná se o technologii Zeroconf, společně se svými známými službami Avahi a Bonjour, které budou využity v průběhu praktické části na více typech operačních systémů. Celá práce je provedena v prostředí IPv6 protokolu pro demonstraci funkčnosti tohoto protokolu se zmíněnými službami.

Teoretická část bakalářské práce je rozdělena do několika kapitol. V první kapitole je rozebrán IPv6 protokol, společně se svou historií a rozdíly oproti staršímu typu IPv4. Zde jsou detailně popsány adresy a adresní prostor tohoto protokolu. Tato kapitola byla přesunuta do přílohy A z důvodu rozsahu práce. Druhá kapitola této bakalářské práce je zaměřena na službu DNS-SD, která je využívána službami Avahi a Bonjur, pro sdílení informací a služeb stanic pomocí DNS záznamů, které jsou zapojeny v lokální síti. Třetí kapitola této práce pojednává o službě mDNS, která je další páteřní službou technologie Zeroconf. V této kapitole jsou rozebrány rozdíly mezi DNS a mDNS, proces dotazování, možnosti tvarů odpovědí, ale hlavně proces této služby a její bezpečnost. Hlavní teoretickou kapitolou této bakalářské práce je kapitola zabývající se technologií Zeroconf, kde jsou popsány principy automatického přidělování Link-Local adresací, průběh mDNS dotazování, proces objevování služeb na síti pomocí DNS-SD a rozdíly mezi službami Avahi a Bonjour. Poslední kapitola se zabývá službami Samba a CIFS pro sdílení souborů přes síť. V této krátké kapitole je základní popis zmíněných služeb, jejich historie, rozdíly a základní technické parametry společně s popisem bezpečnosti.

Důležitým bodem této bakalářské práce je její praktická část. Prvním nejdůležitějším bodem praktické části jsou návrhy implementací výše zmíněných služeb. Celkově byly navrženy tři implementace pro sdílení souborů pomocí služeb Avahi, Bonjour, Samba a CIFS. První implementace je zaměřena na domácí využití. Jedná se o implementaci, která pracuje v prostředí protokolu IPv4 a využívá operační systémy jako Microsoft Windows a GNU/Linux distribuci Raspbian. Na těchto operačních systémech jsou zprovozněny služby Avahi, Bonjour a Samba. Další implementace byla navržena pro využití služby Avahi a služby Samba ve školní laboratoři na GNU/Linux distribuci Ubuntu a Raspbian. Tato implementace byla navržena pro sdílení souborů v prostředí protokolu IPv6. Poslední navrženou implementací je řešení ve virtuálním prostředí, kde jsou všechny služby instalovány na serverovou variantu GNU/Linux distribuce Ubuntu. V poslední variantě byl také využit pouze protokol IPv6 pro sdílení souborů. V praktické části jsou také popsány jednotlivé body v instalaci služeb na různých operačních systémech, zejména je důkladně popsána instalace služby Avahi a služby Samba v GNU/Linux distribuci Ubuntu, ale také instalace služby Bonjour na operačním systému Microsoft Windows. Praktická

část také obsahuje popis nedostatků technologie Zeroconf, ať už to jsou společné nedostatky všech služeb, nebo konkrétní nedostatky jednotlivých služeb. Závěr praktické části je zaměřen na analýzu provozu v jednotlivých implementacích, kde je znovu zmíněn postup implementace, ověření funkčnosti a také jsou zde přiloženy obrázky a grafy například z programu Wireshark, nebo služby tcpdump.

1 DNS Service Discovery

Počet zařízení v TCP/IP neustále stoupá a spousta z těchto zařízení již nejsou klasické pasivní prvky, které čekají na uživatelský vstup a kontrolu. Z větší části se jedná o prvky, nebo zařízení, která můžeme považovat jako plně samostatné prvky TCP/IP sítě. Tyto prvky dokážou samostatně komunikovat s jejich sousedy , nebo plně spoléhat na jiná zařízení v síti, která poskytují služby pro zjednodušení jejich chodu.[1, 2, 3, 4, 5, 6, 7]

Klasická konfigurace síťových parametrů (IP adresace, výchozí brána, směřování, síťová maska apod.) u těchto zařízení může být problematická, vzhledem k jejich nekomfortnímu uživatelskému přístupu. Tento problém je především znát, pokud operátor těchto zařízení nemá dostatečné technické znalosti a není schopen provést konfiguraci správně a dle standartu. Navíc je v celku nepraktické konfigurovat každé zařízení manuálně, vzhledem k rychle narůstajícímu počtu těchto síťových zařízení, proto se vyvíjí způsoby pro automatickou konfiguraci síťových prvků, případně pro automatické sdílení jejich služeb. V posledních letech se trh rozrostl o řadu nových technologií, jak zjednodušit TCP/IP sítě. Ovšem každá automatizace přináší i spoustu nových problémů, na které se musí administrátor sítě připravit.[1, 2, 3, 4, 5, 6, 7]

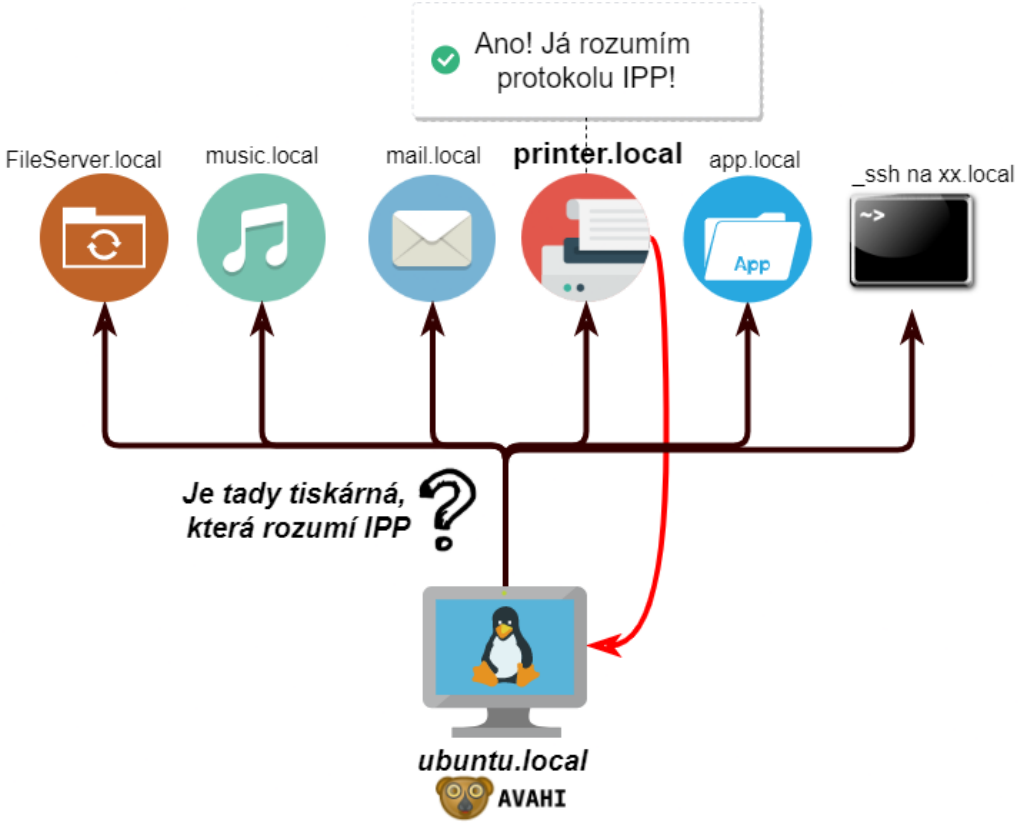
Co se týče automatizace sdílení síťových služeb, můžeme využít například technologii DNS-Service Discovery. Tato technologie je jedena z mnoha způsobů, jak můžeme využít standartní DNS programovatelné rozhraní, servery a pakety k procházení a hledání DNS záznamů v síti. DNS-SD vychází ze staršího AppleTalku, ze kterého převzal jednoduchost, snadné nasazení a využití pro běžné koncové uživatele.[1, 2, 3, 4, 5, 6, 7]

Princip této technologie je popsán v několika síťových standartech RFC, jako například RFC 6760, RFC 6763 a RFC 6887.

1.1 Vyhledávání služeb

Jedním z problémů automatizace sítě je zajistit, aby uživatel nebo síťové zařízení, byli schopni správně najít službu v síti od jednoho nebo více zařízení, a to bez jakéhokoliv nastavení sítě. Důležitý bod tohoto vyhledávání služeb je, že žádné zařízení či uživatel neřeší, od koho nabídka služby dorazila, dokud tato služba má správně vyplněné parametry a je dostupná. Typickým příkladem pro vyhledávání služeb v síti je tisk přes protokol IPP. Uživatel vyšle požadavek do celé lokální sítě, ve kterém hledá IP adresu jakékoliv tiskárny, která dokáže vytisknout dokument pomocí protokolu IPP. Pokud nějaké zařízení v místní síti zná takovou tiskárnu, tak odpovídá uživateli a přeposílá mu IP adresu dané tiskárny. Viz obrázek **1**. [1, 2, 3, 4, 5, 6, 7]

Všechny Service Discovery technologie mají společnou vlastnost a to využívání multicastových adres pro vysílání požadavků a odpovědí. V praxi to pak znamená, že počítač vyšle požadavek (typicky UDP paket) na multicast IP adresu (rozsah 224.0.0.0 až 239.255.255.255 pro IPv4, nebo rozsah ff00::/8 pro IPv6) a tento požadavek je doručen všem stanicím, které naslouchají těmto daným multicastovým adresám, případně konkrétním portům. Viz tabulka **1**.[1, 2, 3, 4, 5, 6, 7]



Obrázek 1: Příklad vyhledávání služeb v lokální síti. [8]

Funkce	**Multicastová skupina**	**IPv4 ekvivalent**
Všichni hosté	*FF02::1*	Broadcast adresa
Všechny směřovače	*FF02::2*	224.0.0.2
OSPFv3 směřovače	*FF02::5*	224.0.0.5
OSPFv3 DR	*FF02::6*	224.0.0.6
RIPng	*FF02::9*	224.0.0.9
EIGRP	*FF02::A*	224.0.0.10
PIM	*FF02::D*	224.0.0.13

Tabulka 1: Tabulka IPv6 multicastových adres.

Celá implementace vyhledávání služeb je postavena takto:

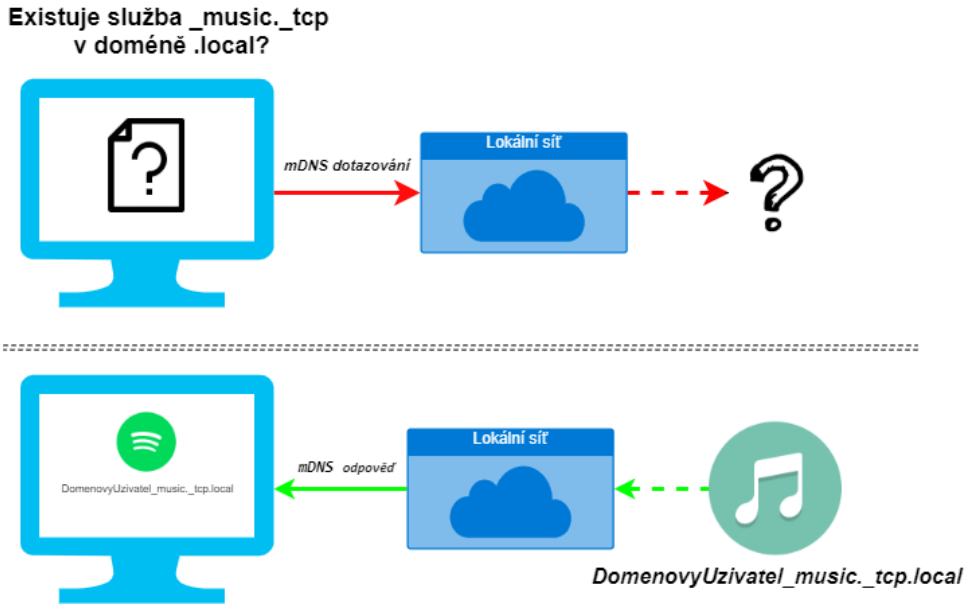
- nově připojená zařízení k síti vysílají informativní zprávu o připojení a svých parametrech,
- aplikace nebo zařízení, které potřebuje konkrétní službu, odešle požadavek popisující požadované vlastnosti služby na konkrétní multicast adresu + konkrétní port,
- všechna zařízení na síti obdrží tento požadavek, a pokud nabízejí shodnou službu, tak odešlou zpět nabídku. Případně pokud znají zařízení, které takovou službu nabízí, předají informaci,
- aplikace nebo zařízení, které požadavek vyslalo, sesbírá všechny odpovědi a z nich si pak vybere poskytovatele služby,
- pokud se jakékoliv zařízení chystá opustit síť, musí ještě před odchodem rozeslat informativní zprávu.

[3, 4, 5]

1.2 Popis služeb

Pokud je v síti vyhledána jakákoliv nová služba, je nutné, aby tato služba byla adekvátně popsána a dodala žadateli potřebné informace. Pokud například budeme využívat aplikaci, která se skládá z více procesů, musíme předem oznámit o jaké procesy se jedná, případně jaké argumenty musí být doplněny pro bezchybný chod dané služby. Ovšem můžou nastat vyjímky, kdy nejsou dodatečné informace potřeba. Zejména v případě, že budeme využívat známé služby (jako například HTTP, IPP, SMB, SSH apod.). U těchto protokolů a služeb nepotřebujeme dodatečné informace, protože k jejich správnému chodu a využití nám stačí pouze nejnutnější informace (IP, port a případně URI), které jsou již předávány v průběhu vyhledávání služeb.[3, 4, 5]
V případě, že k poskytnuté službě nejsou dodatečné informace a není možnost tuto službu provozovat, DNS-SD vrátí pouze označení poskytovatele (jeho adresu), který dodatečné informace může dodat.

Poté co aplikace nebo zařízení obdrží dostatečné množství informací o službě kterou vyhledává, ať už přes samotné vyhledávání služeb nebo od konkrétního zařízení, tak naváže nebo zahodí spojení s touto službou. Tohle už je ale mimo rozsah DNS-SD technologie, o samotné spojení se pak starají protokoly samostatně. Příkladem těchto spojení může být například protokol HTTP (HyperText Transfer Protocol), nebo SOAP, Java RMI(Remote Method Invocation), CORBA(Common Object Request Broker Architecture) a nebo protokoly pro přenos videa a zvuku jako například RTSP (Real Time Streaming Protocol).[3, 4, 5]



Obrázek 2: Přehrávání audio souborů pomocí Media protokolů přes Zeroconf. [**124**]

1.3 Prezentace služeb

Některé technologie Zeroconfu(UPnP, Jini apod.) nabízejí uživatelům grafické rozhraní pro zobrazení služeb (např. centrální správa stanice, správa TV apod.). Tyto rozhraní nabízejí uživateli například přímou správu dané stanice pro rychlejší konfiguraci, nebo případně pro využití dalších služeb a funkcí, které by jinak nebyly dostupné.[3, 4, 5]

To vyžaduje, aby uživatelské rozhraní bylo implementováno způsobem nezávislým na zařízení. Jedním ze způsobů je implementovat toto rozhraní například pomocí HTML (HyperTextový jazyk), které bude sloužit jako vestavěný webový server zabudovaný do zařízení. Dalším z možných způsobů je využití programovacího jazyka Java, který může být spuštěn kdekoliv, kde lze spustit virtuální Java stanici. Grafické rozhraní uživatele pak komunikuje se serverem přes síť konkrétním protokolem.[3, 4, 5]

1.4 Implementace v Zeroconf službách

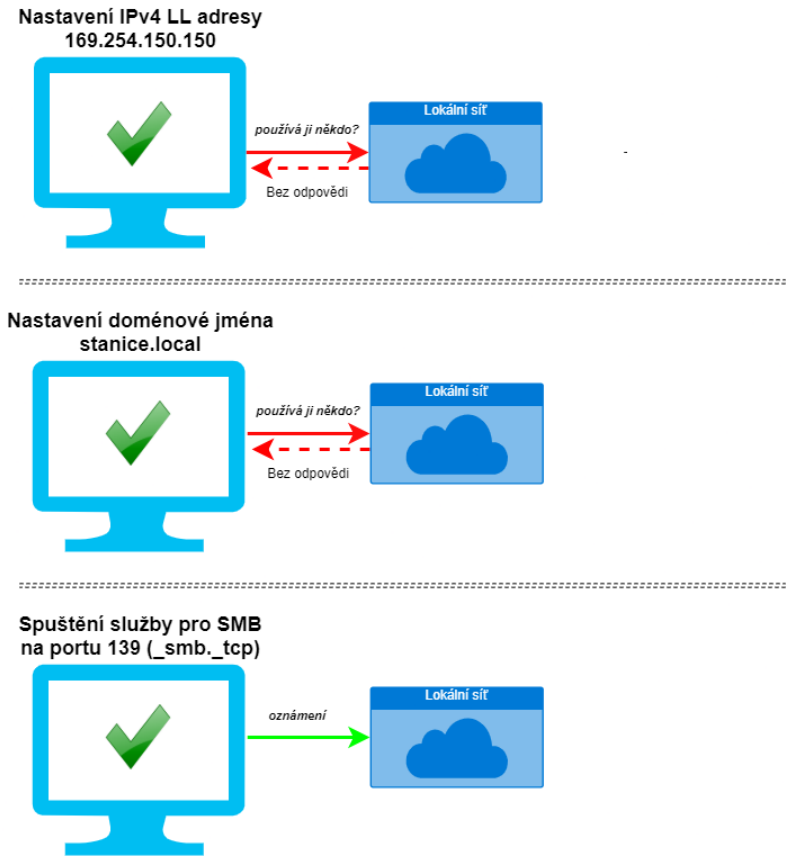
Nejrozšířenější DNS-SD Zeroconf implementace můžeme nalézt v aplikaci Bonjour, nebo v aplikaci Avahi.

- Aplikace Bonjour, od společnosti Apple, byla první implementace Zeroconf protokolu. Implementace má veřejně dostupnou licenci a je zdarma (Open-Source) a může být nasazena na vícero operačních systémech, jako například Mac OSX, Windows, Linux nebo VxWorks. Bonjour je automaticky instalován s operačním systémem Mac OSX a iOS. Instalaci na

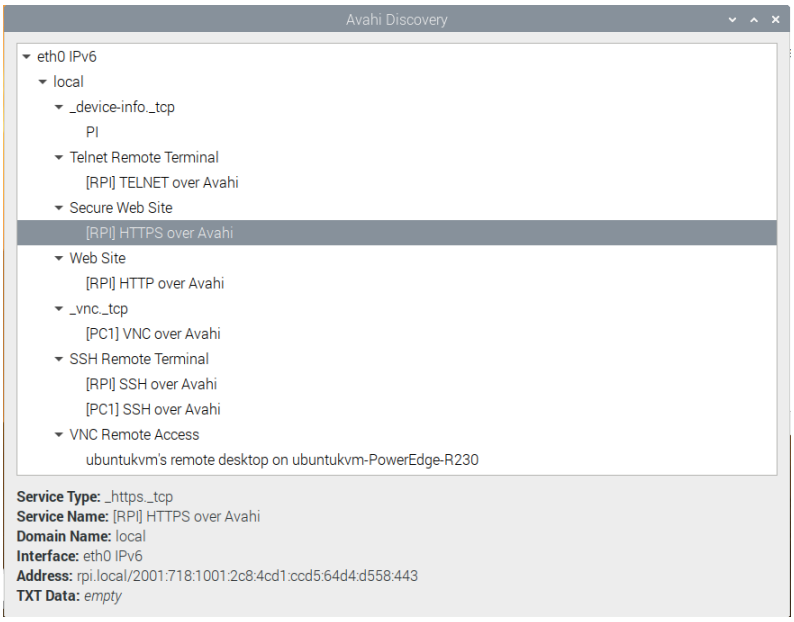
Windows a Linux zajišťuje aplikace iTunes, která má ve své instalaci zahrnutou i instalaci aplikace Bonjour.

- Další variantou je aplikace Avahi, která také implementuje Zeroconf řešení, ale primárně pro operační systémy na bázi Linuxu. Celá aplikace má také veřejně dostupnou licenci a je zdarma (Open-Source). Avahi je automaticky nasazováno do spousty Linuxových distribucí, jako například Debian, Ubuntu, SuSe a další. Aplikaci Avahi je možné nainstalovat také na POSIX platformy jako jsou FreeBSD nebo Solaris.

[3, 4, 5]



(a) Vyhledávání služeb přes aplikaci Bonjour.



(b) Vyhledávání služeb přes aplikaci Avahi (avahi-discover).

Obrázek 3: Vyhledávání služeb přes aplikace Bonjour a Avahi.

2 Multicast DNS

Multicast DNS a jeho doprovodná technologie založená na DNS Service Discovery, která je popsaná v RFC6763, byla vytvořena pro jednoduché nasazení v IP sítích s automatickou konfigurací, která byla známá z AppleTalk technologie (RFC6760).

Multicast DNS si spoustu věcí vypůjčuje již z existujících DNS technologií popsaných v RFC 1034, 1035 a 6195. Využívá již existující strukturu DNS zpráv, syntaxi jmen a typy DNS záznamů.[9, 10, 11, 12, 13, 14, 15]

2.1 Multicastové DNS jména

U stanic, které spravuje jednotlivec či organizace, máme možnost nastavit globální DNS jména a záznamy, jako například "rpi.vsb.cz". V případě, že máme přístup k DNS serveru, tohle řešení je jednoduché a ideální. Nicméně pro většinu uživatelů, kteří vlastní například pouze domácí počítače a jiné stanice, a nemají přístup k DNS serverům, je toto řešení naprosto nereálné a nepraktické. Většina domácích počítačů tak zůstává anonymní pro praktické účely.
Řešení tohoto problému je ve využití mDNS technologie, která umožní stanicím od běžných netechnických uživatelů přidělovat doménové jména pomocí link-local multicastových DNS jmen ve tvaru "rpi-doma.local". Příkladem může být Raspberry PI, které si samo vygeneruje DNS jméno pro lokální síť, ve které také odpovídá na požadavky s tímto jménem. Tento příklad přidělování lokálních doménových jmén můžeme použít na jakémkoliv zařízení v naší síti, nicméně je potřeba dát si pozor na konflikty ve jménech.[9, 10, 11, 12, 13, 14, 15]

Po pojmenování počítače tímto způsobem má uživatel oprávnění pokračovat v používání tohoto jména, dokud nedojde ke konfliktu názvů na síťovém rozhraní, které není vyřešeno ve prospěch uživatele. Pokud k tomu dojde, počítač (nebo jeho lidský uživatel) musí přestat používat toto doménové jméno a měl by se pokusit o přidělení nového jedinečného jména pro použití na tomto síťovém rozhraní. Očekává se, že tyto konflikty budou relativně vzácné u lidí, kteří si vyberou přiměřeně imaginativní jména, ale je stále důležité mít k dispozici mechanismus pro jejich řešení, když k nim dojde.
Technologie mDNS nám v RFC 6762 definuje podmínky užití doménového řádu ".local"a to takové, že tato doména se bere jako doména speciální a dá se využívat pouze v LAN prostředí, nebo-li pouze po místní síti a všechny záznamy z tohoto doménové řádu se berou jako lokální. Tyto podmínky korespondují s adresami, které využíváme pouze pro lokální síť. Jedná se o IPv4 subnet 169.254.X.X/16, nebo IPv6 prefix FE80::/10, kde tyto adresy mají pouze smysl pro lokální rozhraní.
Jakýkoliv DNS dotaz končící na doménové jméno „.local“ musí být směřován na lokální multicastovou adresu 224.0.0.251 mDNS IPv4, nebo její IPv6 ekvivalent FF02::FB . Výběr těchto konkrétních multicastových adres je odůvodněn v RFC 6762 v příloze B ("Appendix B. Design

Rationale for Not Using Hashed Multicast Addresses"). [9, 10, 11, 12, 13, 14, 15]

Není důležité, zda dotaz směřován na jméno končící na „.local“ byl správně doručen, protože uživatel explicitně zadal plně kvalifikovaný název domény, který končí na „.local“. Může se také stát, že uživatel zadá nekvalifikované doménové jméno, ale hostitelský software připojil automaticky lokální doménu, tedy „.local“, protože tato připona se často vyskytovala v historii uživatelského vyhledávání. Tato přípona může být také manuálně nastavena uživatelem jako výchozí pro automatické vyhledávání, případně se může tato doména objevit jako parametr v DHCP (viz RFC 2132), případně v jakémkoliv jiném mechanismu pro automatickou konfiguraci DNS. Nicméně se s touto příponou zachází stejně jako s jakoukoliv jinou doménou, až na rozdíl ve směřovací adrese. [9, 10, 11, 12, 13, 14, 15]

DNS dotazy na jména, která nekončí doménou „.local“ mohou být také odeslány na mDNS multicastovou adresu, pokud není dostupný žádný jiný autoritativní DNS server. Tímto způsobem můžeme uživatelům umožnit stálý provoz komunikace i v případě výpadku sítě. V případě lokální DNS komunikace pomocí mDNS se klade ještě větší důraz na využití DNSSEC (DNS Security Extensions - RFC 4033) zabezpečovací technologie, případně jiného mechanismu, který nám zajistí integritu dat a hlavně ověřuje pravost vzdáleného zařízení.

V případě, že budeme využívat pouze mDNS technologii v naší síti, je doporučeno využívat pouze jednoduché DNS záznamy, jako například "A"pro překlad jmen na IPv4 adresaci, nebo "AAAA"záznamy pro překlad jmen na IPv6 adresaci. Pokud budeme chtít využívat i další typy záznamů, v RFC 6762 je doporučeno využívat souběžně i další DNS technologie, jako například DNS-SD (RFC 6763). [9, 10, 11, 12, 13, 14, 15]

Vyjímka pro unikátní doménové jméno v síti může nastat, pokud budeme chtít například využít load-balancing nebo redundanci konkrétních serverů, nicméně to není doporučeno dle RFC 6762.

2.1.1 Reverzní doménové jména

Jako v případě klasických doménových jmen pro IPv4 a IPv6, mDNS technologie umožňuje využití reverzního mapování záznamů v síti pomocí link-local adres.
Všechny DNS dotazy končící jménem „254.169.in-addr.arpa“ musí být odeslány na link-local mDNS IPv4 multicastovou adresu 224.0.0.251 nebo jeho IPv6 multicast ekvivalent FF02::FB. Tak samo všechny DNS dotazy směřující na reverzní adresu (například „8.e.f.ip6.arpa.“) musí být odeslány na IPv6 link-local mDNS multicast adresu FF02::FB, nebo její IPv4 multicast ekvivalent 224.0.0.251. [9, 10, 11, 12, 13, 14, 15]

2.2 Dotazování

Existují dva způsoby DNS dotazování v rámci mDNS technologie. Jedním ze způsobů je takzvaný jednorázový dotaz, vytvořený pomocí starších překladačů DNS a nepřetržité průběžné dotazy na mDNS vytvořené plně kompatibilními dotazy na mDNS, které podporují asynchronní operace včetně zjišťování služeb založených na DNS. [9]

Ve vzácných případech můžeme natrefit na mDNS responder, který pouze inzeruje sdílené záznamy, nicméně tento mDNS responder musí prvně ověřovat jejich jedinečnost. [9, 10, 11, 12, 13, 14, 15]

2.2.1 Jednorázové mDNS dotazy

Nejzákladnější případ mDNS komunikace spočívá v odesílání dotazů slepě na multicastovou IPv4 adresu 224.0.0.251:5353, případně jeho IPv6 ekvivalent [FF02::FB]:5353, aniž by odesílatel znal tvar multicastové adresy, kterou má použít. Tento způsob DNS komunikace lze jednoduše implementovat. Pokud dotazovaný název odpovídá jakékoliv známé mDNS doméně, pak místo využití nakonfigurované unicast adresace je využita právě multicast IPv4 adresa 224.0.0.251:5353, případně IPv6 ekvivalent [FF02::FB]:5353. Časový limit odpovědi se obvykle zkrátí na dvě nebo tři sekundy. Tyto dotazy jsou většinou prováděny pomocí volných UDP zdrojových portů, ať už dynamicky, nebo staticky. Tyto dotazy však nesmí využívat port 5353, který nám signalizuje přítomnost plně vyhovujícího záznamu.

Příkladem One-Shot mDNS dotazu může být například pokus o otevření webové stránky tiskárny v lokální síti („tiskarna.local“), kdy uživatel obdrží jakoukoliv první odpověď na tento doménový záznam. [9]

2.2.2 Průběžné dotazy

V jednorázových dotazech je základním předpokladem, že transakce začíná, když aplikace vydá dotaz a končí, když je přijata první odpověď. Existuje další typ dotazovací operace, která je více asynchronní, ve které přijetí jedné odpovědi nemusí nutně znamenat, že již nebudou existovat žádné další relevantní odpovědi a operace dotazování pokračuje, dokud nejsou vyžadovány další odpovědi. Určení, kdy nejsou vyžadovány další odpovědi, závisí na typu prováděné operace. Pokud operace vyhledává IPv4 a IPv6 adresy jiných stanic, nejsou po úspěšném připojení k jedné z těchto IPv4 nebo IPv6 stanic vyžadovány další odpovědi. Pokud se jedná o procházení, aby se uživateli zobrazil seznam služeb DNS-SD nalezených v síti, nejsou vyžadovány další odpovědi. Jakmile to uživatel označí softwaru uživatelského rozhraní, např. uzavřením aplikace určené na procházení služeb (Avahi-Discover), které zobrazovalo seznam objevených služeb. [9, 10, 11, 12,

13, 14, 15]

Příkladem těchto průběžných dotazů může být situace, kdy chceme jako uživatel vyhledat všechny dostupné tiskárny v lokální síti. V tomto případě chceme seznam všech dostupných tiskáren, ne jen tiskárnu, která dokázala nejrychleji odpovědět. Většinou toto vyhledávání tiskáren probíhá v nějakém grafickém rozhraní určeném pro tisk, kde se postupně objevují nové dostupné možnosti tisku. Nejlepší varianta pro zobrazení této služby je průběžné mDNS dotazování, protože pokud bychom nastavili předem určený statický seznam tiskáren, nemusí být aktuální (vzhledem k chybovosti zařízení v síti apod.). Proto nejspolehlivější řešení tohoto problému je automatická aktualizace seznamu tiskáren na vyžádání uživatele (pokud uživatel stiskne tlačítko aktualizovat v grafickém rozhraní daného programu). Předpoklady pro toto řešení jsou takové, že interval mezi prvními dvěma dotazy alespoň jedna sekunda, intervaly mezi následnými dotazy musí být zvýšeny nejméně o dvojnásobek. [9, 10, 11, 12, 13, 14, 15]

2.3 Odpovědi

Když mDNS responder vytvoří a odešle zprávu s odpověďmi, oddíly záznamů této zprávy musí obsahovat pouze záznamy, pro které je tento mDNS responder výslovně autoritativní. Tyto odpovědi mohou být generovány, protože záznam odpovídá na otázku obdrženou v mDNS dotazu. Odpovědi na mDNS dotazy nesmí být brány z mezipaměti mDNS responderu, který tyto záznamy získal od jiných mDNS responderů v síti.[9, 10, 11, 12, 13, 14, 15]

Určení, zda daný záznam odpovídá na danou otázku, se provádí pomocí standardních DNS pravidel. Název záznamu se musí shodovat s názvem otázky, záznam rrtyp musí odpovídat otázce qtyp. Pokud qtyp není „ANY“ (255) nebo rrtyp není „CNAME“ (5), a pokud qclass není „ANY“ (255). Stejně jako u unicast DNS se obvykle používá pouze třída DNS 1 („Internet“), ale pokud klientský software používá třídy jiné než 1, musí být použita výše uvedená pravidla shody.[9, 10, 11, 12, 13, 14, 15]

V sítích, které využívají ethernetové médium (IEEE 802.3) a nebo jemu podobné, by měl mDNS responder být schopen zpozdit svou odpověď až o 500 ms. Responder mDNS odpovídá pouze v případě, že má poslat pozitivní nenulovou odpověď nebo autoritativně ví, že konkrétní záznam neexistuje. Odpovědi na mDNS nesmí obsahovat žádné dotazy. Jakékoli dotazy u přijaté odpovědi musí být ignorovány.[9, 10, 11, 12, 13, 14, 15]

2.4 Testování a automatická propagace při startu

Typický mDNS responder by měl obsahovat minimálně tabulku všech svých aktivních rozhraní a záznamů. Tento seznam může být užitečný pro správu daného systému.
Pokaždé když se mDNS responder stanice zapne, probudí ze spánku, nebo zjistí ztrátu konektivity, musí provést základní kroky pro obnovu mDNS procesu. Jedná se o prvotní testování (probing) a následnou propagaci služeb (announcing).

2.4.1 Testování

Prvním krokem po spuštění stanice je testování jedinečnosti služeb na lokální síti. To znamená, že mDNS responder odešle dotaz na všechny své služby (název stanice, IP adresu, apod.) v jednom dotazu a čeká jestli dojde ke konfliktu.
Celý tento proces začíná náhodným odpočtem od 0 ms do 250 ms, kdy se snaží stanice předejít záplavě sítě v případě, že došlo k výpadku na rozbočovači nebo přepínači. Tento proces se poté dvakrát opakuje intervalu v 250 ms. Pokud stanice nenarazí na žádný konflikt, přepne se do druhého kroku propagace služeb.[9, 10, 11, 12, 13, 14, 15]

V případě, že jakákoliv stanice v síti během testování odpoví na tyto dotazy, dochází ke kolizi. Stanice, která vysílá testovací dotazy musí zvolit jiné parametry (název a IP adresaci) a musí znovu proběhnout testovací proces. V případě, že dojde k 15 konfliktům během 10 sekund, pak stanice. která vysílá testovací dotazy, musí počkat 5 sekund, než může znovu vysílat (ochranný mechanismus proti záplavám sítě).[9, 10, 11, 12, 13, 14, 15]

2.4.2 Automatická propagace

Druhým krokem po spuštění stanice je automatické odesílání mDNS záznamů, které jsou nově zaregistrovány, ať už se jedná o unikátní záznamy ze stanice, či sdílené záznamy ze sítě. Pokud je těchto záznamů až příliš a nevlezou se do jednoho mDNS packetu, musí se packetů poslat více.[9, 10, 11, 12, 13, 14, 15]

V případě propagace sdílených služeb (například PTR záznamy) se záznam dává tak, jak je do části packetu, ve které můžeme nalézt odpovědi. V případě propagace záznamů, které byly v předchozím kroku ověřeny jako jedinečné, se umístí do sekce odpovědí s nejvýznamnějším bitem rrclass, který je nastaven na jedna.[9, 10, 11, 12, 13, 14, 15]

V případě odpovědí od mDNS responderu, musí být odeslány ještě minimálně dvě další nevyžádané odpovědi v rozmezí jedné sekundy, aby byla zajištěna zvýšená odolnost proti ztrátě paketů. Celkový počet nevyžádaných mDNS odpovědí může být až osm, za předpokladu, že interval mezi nevyžádanými odpověďmi se při každém odeslání zvětšuje nejméně dvojnásobně.[9,

10, 11, 12, 13, 14, 15]

2.5 Řešení konfliktů

Ke konfliktu dochází, když má mDNS responder jedinečný záznam, pro který je aktuálně autoritativní a obdrží zprávu s odpovědí obsahující záznam se stejným názvem, rrtypem a rrclassem, ale nekonzistentní rdata. Běžným příkladem typu záznamu, který má být jedinečný a ne sdílený mezi hostiteli, je záznam adresy, který mapuje jméno hostitele na jeho IP adresu. Pokud jiné zařízení v síti rozešle odpověď se stejným názvem stanice, ale jinou IP adresou, považuje se taková odpověď za konfliktní.[9, 10, 11, 12, 13, 14, 15]

Pokud stanice obdrží konfliktní mDNS odpověď, musí se automaticky přepnout zpět do testovacího kroku a vyresetovat všechny své parametry. Poté se kontrolují shodné parametry (například jméno stanice) a ověřuje se, která z těchto stanic má své jméno nastaveno staticky, a která pouze vygenerované.[9, 10, 11, 12, 13, 14, 15]

Stanice s konfliktem by měly projít tyto doporučené kroky:

- zajistit nové unikátní jméno stanice jakýmkoliv způsobem, například automatické vygenerování jména,
- po nově přiděleném jménu musí stanice vysílat testovací mDNS dotazy na své jméno a čeká se, jestli náhodou nedojde k dalšímu konfliktu,
- pokud nedojde při testování ke konfliktu, předpokládá se, že vybrané jméno je unikátní a stanice si toto jméno zapíše do svých konfiguračních souborů,
- na stanici, kde došlo ke změně názvu, musí být informován její uživatel pomocí systémových zpráv, případně zpráv v aplikaci, která využívá mDNS služby. (V případě serverové verze stanice se tato zpráva může zapsat do loggu zařízení, nebo vyslat SNMP upozornění.)

[9, 10, 11, 12, 13, 14, 15]

2.6 IPv6 kompatibilita

Stanice, které pracují buďto pouze s IPv4, nebo IPv6 se navzájem nevidí, ani když jsou na stejném ethernet médiu, nevšímají si vzájemného provozu. Z tohoto důvodu může mít každý fyzický propoj dvě nesouvisející ".local"zóny, jedna určená pro IPv4 a druhá pro IPv6.[9, 10, 11, 12, 13, 14, 15]

Stanice využívající dual-stack (v4 / v6) se mohou účastnit obou „.local“ zón, a měli by zaregistrovat své jméno (názvy) a provádět svá vyhledávání pomocí IPv4 a IPv6 současně. To jim

umožňuje připojení ke stanicím, které využívají buďto pouze IPv4, nebo IPv6. Ve výsledku to má takový efekt, že stanice s dual-stackem se tváří jako jedna fyzická jednotka s oddělenými logickými jednotkami pro IPv4 a IPv6. Každá taková stanice pak generuje NSEC záznamy, které naznačují, že název stanice obsahuje jak A tak AAAA záznamy.[9, 10, 11, 12, 13, 14, 15]

2.7 Bezpečnost

Algoritmus pro detekci a řešení konfliktů jmen je ze své podstaty algoritmus, který předpokládá spolupracující účastníky. Jeho účelem je umožnit skupině stanic dospět k vzájemně nesouvislé sadě názvů, pokud neexistuje žádná centrální autorita (DNS server), která by toto přidělování koordinovala nebo řešila spory. Při absenci DNS serveru k řešení sporů je jedinou alternativou to, že stanice musí spolupracovat společně, aby dospěly k řešení.[9, 10, 14]

V prostředí, ve kterém jsou účastníci vzájemně antagonističtí a nechtějí spolupracovat, jsou vhodné jiné mechanismy, jako je ručně nakonfigurovaný DNS server, případně lokálně uložené DNS záznamy v jednotlivých stanicích.[9, 10, 14]

V prostředí, kde existuje skupina spolupracujících stanic, ale není jistota, že na stejném fyzickém připojení není připojen útočník, či nespolupracující stanice, musí tyto stanice využívat zabezpečovací služby jako IPsec, nebo DNSSEC (RFC4033), aby mohly správně rozlišovat validní mDNS dotazy důvěryhodných účastníků od těch falešných (podvodných) dotazů, které poté zahodí.[9, 10, 14]

V případě globálních DNS dotazů, které mohou nastat při výpadku WAN konektivity, nebo při klasických DNS dotazech, je důležité využívat DNSSEC službu, protože stanice může dostat falešné odpovědi z lokální sítě od útočníka, který se maskuje globálními DNS jmény.[9, 10, 14]

Většina uživatelů zanedbává psaní plně kvalifikovaného názvu domény, což z něj činí relativní název domény (příkladem může být „www.vsb.cz“). V případě výpadku sítě se pokusy o překlad této adresy nezdaří, což povede k použití lokálního seznamu hledání, včetně mDNS domény „.local“, pokud existuje. Útočník se v tomto případě dokáže maskovat jako uživatelem vyhledávaná doména („www.vsb.cz“) a může pomocí mDNS odpovědět na tento dotaz pomocí „www.vsb.cz.local“. Aby se tomu zabránilo, stanice nesmí připojovat vyhledávací suffix „.local“ k jakémukoliv relativnímu, i částečně kvalifikovanému, DNS záznamu.[9, 10, 14]

3 Zeroconf

Za zrodem Zeroconf technologie stojí Stuart Cheshire, který se touto problematikou zabýval již v roce 1997. O pár let později (1999) organizace IETF uspořádala dvě setkání „Birds of a Feather“ (BOF) na březnových a červencových schůzích IETF na téma „Networking in the Small“ (NITS), kterým předsedal Stuart Cheshire a Peter Ford. V rámci setkání NITS BOF byla v září 1999 vytvořena pracovní skupina Zero Configuration Networking (Zeroconf). V květnu 2002 Apple představil svou vlastní verzi zeroconf aplikace a ochranou známku pod názvem „Rendezvous“. Bohužel pro společnost Apple měla jiná společnost také síťový produkt s názvem „Rendezvous“ a v dubnu 2005 společnost Apple oznámila nový název své technologie Zeroconf: „Bonjour“. Název a logo Bonjour mohou mít i jiné produkty třetích stran. Mezitím byly také vytvořeny další Open-Source implementace technologií Zeroconf, včetně Howl a Avahi.[10, 16, 17]

Na technické úrovni je Zeroconf kombinací tří technologií. Co je však důležitější, je to, že uživatel nemusí mít žádné technické predispozice pro nastavení Zeroconf produktů, tyto produkty „prostě fungují“. Nastavení síťového zařízení by mělo být stejně snadné jako nastavení nové stolní lampy - zapojíte ji, zapnete ji a bude fungovat. Znamená to tedy, že běžný uživatel je schopen pracovat na stanici, aniž by musel znát správné nastavení pro například DHCP, nebo DNS.[10, 16, 17]

Zeroconf je tedy o dvou věcech:

- vyrábět produkty, které se opravdu snadno používají,
- vytváření podpůrných technologií, které toto umožňují.

[10, 16, 17]

Celá myšlenka Zeroconfu stojí na třech základních technologiích, jedná se o:

- automatické přidělování link-local IP adres na rozhraní pomocí AIPA / APIPA (Automatic Private IP Addressing) ,
- odesílání dotazů po místní síti přes multicast DNS (mDNS),
- objevování služeb sdílených po síti pomocí DNS Service Discovery.

[10, 16, 17]

3.1 Automatické přidělování link-local IP adres

Aby bylo možné provádět jakoukoli síťovou komunikaci, potřebuje počítač IP adresu a většina počítačů ji dnes obvykle obdrží pomocí DHCP. DHCP je dokonalý protokol a automatické přidělování link-local adres s ním nekonkuruje. Adresace na rozhraní pomocí link-local adres by

se dala spíše považovat jako záchranná síť. Pokud DHCP server spadne nebo není k dispozici, link-local adresace umožňuje počítači vytvořit automatickou adresaci, se kterou dokáže i nadále komunikovat, alespoň na lokální síti, i když není možná širší komunikace. V případě zařízení, které nemá obrazovku nebo klávesnici a je konfigurováno výhradně vzdáleně pomocí sítě, je toto záložní link-local adresování obzvláště důležité. Pokud by se toto vzdálené zařízení dostalo do stavu, kdy ztratilo věskerou možnost komunikace, a to i v místní síti, pak by neexistoval jiný způsob, jak s daným zařízením komunikovat a opravit případnou chybu v konfiguraci, která nastala.[10, 16, 17, 18, 19]

V případě přidělení adres pomocí DHCP nebo manuálního nastavení, se předpokládá určitý druh centrálního orgánu pro dohled nad přidělováním IP adres. V případě využití DHCP serveru, se touto centrální autoritou stává on a kontroluje přidělování adres. V případě manuálního nastavení adresace na stanici se předpokládá, že osoba která provádí toto nastavení je k tomu oprávněna. U aplikace Zeroconf se výběr adres provádí distribuovaným způsobem. Každé zařízení je zodpovědné za výběr své vlastní adresy a za ověření, zda může použít vybranou adresu.[10, 16, 17, 18, 19]

Pro automatickou link-local adresaci je rezervován adresní prostor 169.254.0.0/16, kde prvních a posledních 256 adres bylo rezervováno pro budoucí užití, zbylých 65 654 adres je volných pro link-local použití. Link-local adresace není určená pro rozlehlé sítě s například 65 000 zařízeními, v takovém případě je potřeba využít DHCP servery a dedikovaného člověka jako správce sítě. Využití technologie Zeroconf s link-local adresací je primárně určeno pro malé lokální adhoc sítě, kde je požadovaná komunikace i bez centrálního DHCP serveru (v případě výpadku) a zároveň je pro tyto sítě zajištěno alespoň minimální zabezpečení.
Protože se v tomto scénáři nepočítá s žádným DHCP serverem, zařízení které se chce připojit do takové sítě musí zvládat případné řešení konfliktů. Primárně je tedy tato technologie určena pro sítě o 2, 10, nebo i 100 zařízeními, ačkoliv analýza v RFC 3927 ukazuje, že sítě s více než 1 000 zařízeními jsou stále přiměřeně dobře funkční. Zařízení, které se připojuje do tak velké sítě má stále 98% šanci na to, že si zvolí volnou IPv6 LL adresu a nedojde tak k žádnému konfliktu. V případě konfliktu je šance 99.96%, že druhá volba adresy již bude validní. Šance na více než 10 konfliktů během volby IPv4 LL adresy je jedna ku 10^{17} .[20, 10, 16, 17, 18, 19]

Na rozdíl od toho se IPv6 LL adresy používají zároveň s ostatními IPv6 adresami. Obvykle je tato adresa přiřazena na rozhraní hned po inicializaci tohoto rozhraní. Celý popis generace této IPv6 adresy je popsán v RFC 4862.[21, 20]

IPv6 LL adresa je vytvořena pomocí dobře známého prefixu FE80::0 (RFC 4291) a identifikátoru rozhraní (MAC adresa) následujícím způsobem:

1. bity nejvíce nalevo z délky prefixu adresy jsou prefixem link-local adresy,

2. bity v pravo za link-local prefixem jsou vynulovány,

3. pokud je délka identifikátoru rozhraní N bitů, jsou N bity z práve strany nahrazovány indetifikátorem rozhraním.

[20, 10, 16, 17, 18, 19]

Pokud je součet link-local prefixu a N větší než 128, automatická konfigurace selže a je vyžadována ruční konfigurace. Délka identifikátoru rozhraní je definována v samostatném dokumentu specifickém pro typ propojení, který by měl být také v souladu s adresovou architekturou viz RFC 4291.

3.2 mDNS dotazy

Jak již bylo popsáno v předešlé kapitole zabývající se mDNS technologií, pokud neexistuje žádný dostupný DNS server, případně pokud existuje, ale nemáte žádnou možnost jak do něj přidat vlastní doménové záznamy, služba mDNS Vám poskytne způsob, jak jednoduše vytvořit doménové záznamy a odkazovat se na zařízení v místní síti dle jeho názvu. Pokud tedy není žádné autoritativní zařízení určené pro DHCP nebo DNS dostupné, na jeho místě jej nahradí mDNS technologie, která je schopná navazovat TCP spojení na lokální síti s využitím link-local adresace. Tímto ale Zeroconf nekončí, s dosud popsanými technologiemi jsme mohli provádět základní, ale užitečné, nastavení sítě, ale pokud tyto technologie chcete využívat, musíte znát lokální názvy stanic. V případě, že při využívání služeb špatně zadáte toto lokální doménové jméno, nebo jej zapomenete, pravděpodobně nebudete schopni navázat spojení ani v lokální síti. Proto se v návaznosti na mDNS využívá také technologie DNS Service Discovery, která Vám umožní vyhledávat předem všechny známé služby a zařízení.[10, 16, 17, 18, 19]

Ale proč vlastně používat doménová jména v síti? Proč například nepoužívat jen IP adresy? IP adresace se v průběhu času může změnit, pokud nenastavujeme všechna zařízení v síti staticky, ale hlavně lidé mají problém pamatovat si místo názvu zařízení IP adresu. Proto je důležité využívat lokálně unikátní doménová jména, která se budou jednoduše pamatovat, ale hlavně je neafektuje změna adresace. V případě, že chceme využívat také, nebo pouze IPv6 adresaci, je obzvlášť důležité nastavit unikátní doménová jména, protože tyto adresy jsou již opravdu těžce zapamatovatelné.[10, 16, 17, 18, 19]

3.3 Objevování služeb DNS-SD

Další z technologií, která je klíčová pro Zeroconf službu, je DNS Service Discovery, která nám pomáhá vyhledávat služby v lokální síti. Celá tato technologie je popsána výše v druhé kapitole.[10, 16, 17, 18, 19]

Ve světě síťových zařízení není možné říci, že existuje zařízení, se kterým nelze komunikovat. Obvykle antropomorfizujeme zařízení způsobem, který není zcela správný. Říkáme, že jsme „pingovali server“, ale ve skutečnosti to, co jsme pingovali, byl kus softwaru na serveru, který odpovídá paketům ICMP echo. Pokud tento software odeberete, přestane odpovídat na požadavky na ping, přestože je server stále k dispozici a může stále dobře plnit další funkce. Při návrhu systému pro Service Discovery službu je důležité si uvědomit, že to, co klienti využívající síť chtějí objevit, jsou softwarové entity, se kterými mohou komunikovat, nikoli pouhé kusy hardwaru. Rozdíl mezi objevováním pouhých služeb a objevováním hardwaru se může zdát malý a nedůležitý, ale ve skutečnosti je markantní. Uživatelé v grafickém rozhraní chtějí vidět reálný seznam služeb a zařízení, na kterých mohou například tisknout dokumenty. V iTunes chtějí vidět seznam zařízení podporujících přehrávání hudebních zdrojů. V prohlížeči fotek chtějí vidět dostupná fotoalba z různých zařízení na síti. Nebo například ve webovém prohlížeči chtějí uživatelé vidět seznam nabízených webových stránek, kterými mohou brouzdat. Jakýkoliv hardware na síti nemusí nabízet žádnou z těchto služeb, nebo jich může nabízet hned několik. Díky technologii Service Discovery vidíme seznam služeb v síti, které můžeme použít, nikoliv seznam hardwaru, na kterém mohou, nebo nemusí tyto služby být.[10, 16, 17, 18, 19]

3.4 Avahi

Avahi je bezplatná implementace Zeroconfu, včetně systému pro vyhledávání služeb mDNS a DNS-SD. Umožňuje programům a stanicím publikovat a objevovat služby a jiné stanice v lokální síti bez jakékoliv specifické konfigurace. Avahi je licencováno na základně GNU (LGPL - Lesser General Public License) licence.[22, 23, 24, 25, 10, 26, 27, 28]

Umožňuje programům publikovat a objevovat služby a hostitele běžící v místní síti bez specifické konfigurace. Můžete se například připojit k síti a okamžitě najít tiskárny, na kterých se má tisknout, soubory, na které se mají dívat a uživatele, se kterými by se měly bavit.[22, 23, 24, 25, 10, 26]

Celá tato implementace Zeroconfu se dělí na vícero podprocesů, které jsou pro běžného uživatele neviditelné.

3.4.1 Avahi Daemon

Jedná se o hlavní podproces této implementace, Avahi Daemon registruje lokální IP adresy a statické služby pomocí mDNS a DNS-SD, také poskytuje dvě IPC API pro lokální programy, které jej využívají pro zápis dat do mDNS lokální cache paměti. Prvním krokem je takzvaný "jednoduchý protokol", který je exkluzivně využíván procesem Avahi-Dnsconfd. Jedná se o další Avahi Daemon, který konfiguruje unicast DNS servery pomocí informací obdržených z mDNS. Druhým krokem je využití repozitáře nss-mdns, jedná se o libc NSS plugin, který překládá adresy z mDNS komunikace. Finálním krokem je povolení rozhraní D-Bus, které poskytuje bohaté objektově orientované rozhraní pro aplikace podporující D-Bus.[22, 23, 24, 25, 10, 26, 27, 28]

Ihned po zapnutí stanice, která obsahuje Zeroconf implementaci Avahi, začne tato aplikace číst svou konfiguraci uloženou v souboru:

```
/etc/avahi/avahi-daemon.conf
```

poté přejde na čtení XML fragmentů, které mohou definovat DNS-SD služby. Tyto fragmenty se ukládají zde:

```
/etc/avahi/services/*.service
```

kde * vyjadřuje název fragmentu - například "_ssh."

3.4.2 Avahi Browse

V tomto podprocesu Avahi vyhledává dostupné mDNS a DNS-SD služby v lokální síti s využitím Avahi Daemonu.[22, 23, 24, 25, 10, 26, 27, 28]

3.4.3 Avahi Publish

V tomto podprocesu Avahi registruje své mDNS a DNS-SD služby (doménové jména a IP adresy) v lokální síti s využitím Avahi Daemonu.[22, 23, 24, 25, 10, 26, 27, 28]

3.4.4 Avahi Resolve

Tento podproces slouží pro manuální překlad doménových jmen na IP adresy (a opačně) s využitím Avahi Daemonu.[22, 23, 24, 25, 10, 26, 27, 28]

3.4.5 Avahi Set-Host-Name

Pomocí tohoto podprocesu služba Avahi registruje aktuální jméno zařízení do mDNS služby, která jej pak prezentuje v síti. Účinek této operace není trvalý, kvůli restartům Avahi Daemona. Tato operace je obvykle privilegovaná.[22, 23, 24, 25, 10, 26, 27, 28]

3.4.6 Avahi Hosts

Avahi Hosts není podproces jako takový, jedná se spíše o soubor, který lze využít pro statické definování DNS záznamů v lokální síti (překlad jmen na IP adresy). To je obzvláště užitečné při publikování DNS-SD služeb jménem jiných hostitelů. Tento soubor nalezneme zde:

```
/etc/avahi/hosts
```

Formát souboru je podobný formátu souboru:

```
/etc/hosts
```

Na každém řádku můžeme zapsat IP adresu a odpovídající název hostitele. Názvy hostitelů by měly být ve formě FQDN, to znamená s připojenou příponou .local.[22, 23, 24, 25, 10, 26, 27, 28]

3.5 Bonjour

Bonjour od společnosti Apple je další implementace Zeroconfu. Bonjour vyhledá zařízení, jako jsou tiskárny, počítače a služby, které tato zařízení nabízejí v lokální síti pomocí mDNS. Tento software je dodáván s operačními systémy MacOS a iOS společnosti Apple. Bonjour lze také nainstalovat do počítačů se systémem Microsoft Windows. Komponenty Bonjour mohou být také zahrnuty do jiného softwaru, jako jsou iTunes a Safari. Po zavedení v roce 2002 s Mac OS X 10.2, pod názvem Rendezvous, byl software přejmenován v roce 2005 na Bonjour. [29, 10, 30, 31, 32, 19, 33, 34]

Tento software je široce používán v celém systému MacOS a umožňuje uživatelům nastavit síť bez jakékoli konfigurace. Od roku 2010 se používá k nalezení tiskáren a serverů pro sdílení souborů.[29, 10, 30, 31, 32, 19, 33, 34]

Mezi významné aplikace využívající Bonjour patří:

- iTunes - pro hledání sdílené hudby,
- iPhoto - pro hledání sdílených fotek,
- iChat, Adobe Systems Creative Suite 3, Proteus, Adium, Fire, Pidgin, Skype, Vine Server a Elgato EyeTV pro komunikaci s více klienty,
- Gizmo5 - pro hledání klientů na lokální síti,
- SolidWorks - pro správu licencí,
- Things a OmniFocus - pro synchronizaci projektů mezi Apple zařízeními,
- Safari - pro vyhledání webových serverů na lokální síti.

[29, 10, 30, 31, 32, 19, 33, 34]

Software jako Bonjour Browser nebo iStumbler, oba pro MacOS, lze použít k zobrazení všech Bonjour služeb. Aplikace Apple „Remote“ pro iPhone a iPod Touch používá Bonjour k navázání připojení ke knihovnám iTunes prostřednictvím Wi-Fi.

Bonjour funguje pouze v rámci jedné broadcastové domény, což je obvykle malá oblast, bez zvláštní konfigurace DNS. MacOS, Bonjour pro Windows a AirPort Base Stations mohou být nakonfigurovány pro použití Wide Area Bonjour, který umožňuje využití mDNS přes WAN prostřednictvím vhodně nakonfigurovaného serveru DNS. [29, 10, 30, 31, 32, 19, 33, 34]

Aplikace obecně implementují službu Bonjour pomocí klasických TCP/IP konexí nikoli v samotném operařním systému. Ačkoliv macOS poskytuje různé Bonjour služby, Bonjour pracuje i na jiných operačních systémech. Společnost Apple zpřístupnila zdrojový kód Bonjour mDNS responderu a Bonjour DNS-SD, jako projekt Darwin open source. Projekt poskytuje zdrojový kód na sestavení Zeroconf Daemona pro širokou škálu platforem včetně MacOS 9, MacOS, Linux, BSD, Solaris, VxWorks a Windows. Apple také poskytuje uživatelům operačního systému Microsoft Windows instalační soubor s názvem Bonjour pro Windows. [29, 10, 30, 31, 32, 19, 33, 34]

3.5.1 Licence

Bonjour je volně dostupný jako FreeWare software. Pokud jej ale chtějí vývojáři používat, může se stát, že musí dostat zvláštní povolení, ale většinou stačí použít Bonjour logo v projektu. Zdrojové kódy pro mDNS responder jsou k dispozici pod Apache licencí. [29, 10, 30, 31, 32, 19, 33, 34]

3.5.2 Bonjour a Windows

Bonjour verze 2.0, vydaný 24. února 2010, funguje se systémy Microsoft Windows 2000, 2003, XP, Vista, 7, 8 a 10. Systémy jej využívají především k usnadnění instalace, konfigurace a používání síťových tiskáren a běží tedy od spuštění. Bonjour pro Windows také přidává možnosti Zeroconf do aplikace Internet Explorer. [29, 10, 30, 31, 32, 19, 33, 34]

Instalační soubor v systému Windows obvykle ukládá Bonjour data do složky nazvané „Bonjour“ ve složce „Program Files“. Bonjour během své instalace upravuje systémové registry související s konfigurací a provozen lokální sítě. Po své instalaci Bonjour spouští mDNSResponder.exe, který spustí komunikaci v síti na portu UDP 5353. Využití tohoto portu v operačním systému Windows může vyžadovat dodatečnou konfiguraci Firewallu, pokud se jedná o korporátní síť. Kompletní instalace Bonjour v operačním systému Windows instaluje dodatečnou službu pro Internet Explorer, průvodce tiskáren a dodatečnou konfiguraci síťových služeb. Dodatečné Bon-

jour služby mohou být nainstalovány prostřednictvím dalšího Apple softwaru, jako je například iTunes. [29, 10, 30, 31, 32, 19, 33, 34]

3.6 UPnP

(Universal Plug aNd Play) Skupina protokolů, které umožňují přenos dat mezi počítači, mobilními zařízeními a A/V zařízeními. Technologie UPnP, představená v roce 1999 fórem UPnP (www.upnp.org), poskytuje automatické vyhledávání zařízení v TCP/IP síti. Je široce používaná pro streamování hudby a videí do zařízení (DLNA). UPnP podporuje QoS, proto se také používá v bezpečnostních kamerách, osvětlení a klimatizacích.[35, 36, 37]

UPnP technologie je podobná Zeroconf technologii, obě technologie poskytují vyhledávání služeb a automatické přidělení IP adres, ale využívají různé protokoly.[35, 36, 37]

UPnP se využívá pro automatické otevíraní portů na směřovačích, aby lokální síť byla dostupná z internetu. Příkladem může být otevření portů pro přenos zvuku a videa obousměrně tak, aby bez ohledu na to, která strana zahájí hovor, provoz prošel bez problému.[35, 36, 37]

4 Samba a CIFS

Common Internet File System (CIFS) je síťový souborový protokol používaný k poskytování sdíleného přístupu k souborům a tiskárnám mezi stroji v síti. Klientská aplikace CIFS umí číst, psát, upravovat a dokonce odstraňovat soubory na vzdáleném serveru. Klient, využívající CIFS protokol, může komunikovat s jakýmkoli serverem, který umí číst požadavky tohoto protokolu. [38, 39, 40, 41, 42, 43, 44, 45, 46, 47] Aktuální implementace CIFS protokolu od společnosti Microsoft jsou de facto standarty tohoto protokolu. Protokol byl vyvinut v roce 1980 Barry Feigenbaumem v IBM. Později byl tento protokol také známý jako SMB (Server Message Block). SMB byl původně navržen tak, aby fungoval nad rozhraním NETBIOS / NETBEUI API (obvykle implementovaným s NBF, NetBIOS přes IPX / SPX nebo NBT) s cílem vyladit přístup k místním souborům přes síť.[38, 39, 40, 41, 42, 43, 44, 45, 46, 47]

S vydáním systému Windows 95 na začátku 90. let společnost Microsoft provedla značné úpravy nejběžněji používané verze SMB. Microsoft poté sloučil aktualizovanou verzi protokolu SMB (CIFS) s aplikací LAN Manager, která přináší podporu klienta i serveru. Díky tomu mohou uživatelé vytvářet specifické požadavky na server a ten poté reaguje dle potřeby uživatelů. Data jsou vyměňovány pouze mezi ověřenými počítači a servery. V průběhu let vzniklo několik variant a modelů využití tohoto protokolu, kde původně sice sloužil pouze ke sdílení souborů, ale s rostoucími sítěmi vznikly další požadavky na nové síťové služby, oddělení oprávnění a celkovému rozsahu využití CIFS.[38, 39, 40, 41, 42, 43, 44, 45, 46, 47]

Postupem času byl protokol CIFS využíván na operačních systémech různě, příkladem může být:

- systém DOS - v tomto případě protokol CIFS byl využit pouze pro jednoduché sdílení souborů v NETBIOS prostředí. Fungovalo to tak, že se předem určil seznam adresářů, které mají být sdíleny a jejich návzy, a poté se nastavilo heslo pro sdílení,
- Windows NT - na operačním systému Windows NT byl vyvinut první koncept seskupování počítačů a serverů do domén. Byly zavedeny první termíny členství v doméně, jednalo se například o DC (řadič domény) a PDC (primární řadič domény),
- Windows od verze 2000 - od verze 2000 byla v operačním systému Windows zavedena podpora pro AD (Active Directory), které bylo implementováno pomocí modifikovaných protokolů. Pojmem "Aktivní doména"je myšleno, že nyní existují různé ústřední body pro udělování oprávnění počítačům a uživatelů, ale hlavně omezení velikosti domény se zmenšilo (příklad je využití více domén naráz). Postupem času se protokol SMB/CIFS průběžně aktualizoval,

- Windows 2000, XP a Server 2003 - v těchto OS můžele nalézt první verzi SMB a CIFS (SMB1.0),
- Windows Vista, Server 2008 - v dalších OS přichází nová verze SMB a to verze 2.0, která zvýšila rozsah pro sdílení souborů, zvýšila výkon sdružování požadavků, ale hlavně zvýšila rychlost čtení a zápisu. Protokol se také stal robustnějším, ale bezpečnějším, šifrování bylo změněno z MD5 na HMACSHA-256,
- Windows 7, Server 2008 R2 - následně byl vylepšen na verzi SMB2.1, která podporovala větší MTU, Branch Cache a také vylepšení v oblasti pronájmu souborů,
- Windows 8, Server 2012 - verze SMB3.0 umožňuje multikanálové vysílání, SMBDirect, ale hlavním přínosem bylo lepší řešení v případě selhání služby, které zlepšilo výkon a škálovatelnost. Verze 3.0 se také rozrostla o zálohování, nové zabezpečení, správu pro SQL server a PowerShell. V oblasti zabezpečení zavádí end-to-end šifrování a nový algoritmus podpisů založený na AES,
- Windows 8.1, Server 2012 R2 - v těchto operačních systémech byla implementována verze SMB3.02, zavedla možnost zcela zakázat staré verze SMB (CIFS/SMB1.0), včetně odstranění starých binárních souborů. Tato konfigurace není výchozí, ale je silně doporučována Microsoftem, hlavně v případě, kdy využíváme například Hyper-V virtualizaci,
- Windows 10, Server 2016 a 2019 - v nejnovějších operačních systémech od společnosti Microsoft můžeme nalézt verzi SMB3.1.1, která opět rozšířila zabezpečení o nové šifrování, tentokrát přinesla AES-128-GCM (místo staršího AES-128-CCM), ale hlavně implementuje kontrolu integrity před autentizací pomocí SHA-512. SMB3.1.1 také vyžaduje povinné vyjednávání při připojení ke klientům využívající verze SMB2.X a vyšší.

[38, 39, 40, 41, 42, 43, 44, 45, 46, 47]

4.0.1 Samba

Většina lidí v podstatě ani neví, jestli používají SMB, nebo CIFS protokol. Oba jsou vzájemně zaměnitelné nejen v popisu, ale i v samotné aplikaci. V praxi to znamená, že klient využívající protokol CIFS, může navázat TCP/IP spojení se serverem, který používá protokol SMB a naopak. Důvodem je, že v podstatě protokol CIFS je forma SMB, ale ikdyž jsou to téměř totožné protokoly, stále existují markantní rozdíly v implementaci, ale především v ladění výkonu (proto se využívají různé názvy). Markantní rozdíly těchto protokolů můžeme především sledovat ve výkonu přes LAN a WAN sítě, hromadné úpravy souborů či jejich čtení.[38, 39, 40, 41, 42, 43, 44, 45, 46, 47]

Implementace pouze protokolu CIFS se v dnešní době využívá již zřídka. Většina moderních počítačů (2005+) s úložným systémem používá SMB2, nebo SMB3. Ve světě Windows je SMB2

standartem od Windows Vista (2006) a SMB3 od Windows 8 / Server 2012.

CIFS má mezi odborníky negativní konotaci. SMB 2 a SMB 3 jsou masivní vylepšení nad dialektem CIFS a architekti úložišť, kteří jsou blízko a mají rádi protokoly pro sdílení souborů, neuznávají chybný název. Je to něco jako pověřit výkonného asistenta sekretářkou.[38, 39, 40, 41, 42, 43, 44, 45, 46, 47]

5 Realizace praktické části

5.1 Návrhy realizací

Návrh realizace praktické části spočíval ve vícero síťových zapojení, ve kterých bylo možné otestovat technologii Zeroconf, ať už se jednalo o službu Avahi, nebo Bonjour.

Jedna z prvních testovacích implementací probíhala v domácím prostředí, kde bylo cílem vyzkoušet funkčnost buďto samotné služby Bonjour, nebo kooperace služby Avahi a Bonjour najednou. V této implementaci bylo využito již zapojené a zaběhlé sítě, která využívá spousty různorodých síťových prvků. Více informací k této implementaci je v kapitole **5.1.1**.

Další implementace probíhali ve školní laboratoři POREB215 a ve virtuálním prostředí zprostředkovaného serverem. Tyto implementace využívali pouze službu Avahi, a to na operačních systémech Ubuntu, Ubuntu Server a Raspbian. Informace k těmto impelemntacím jsou uvedeny v kapitole **5.1.2** a **5.1.3** .

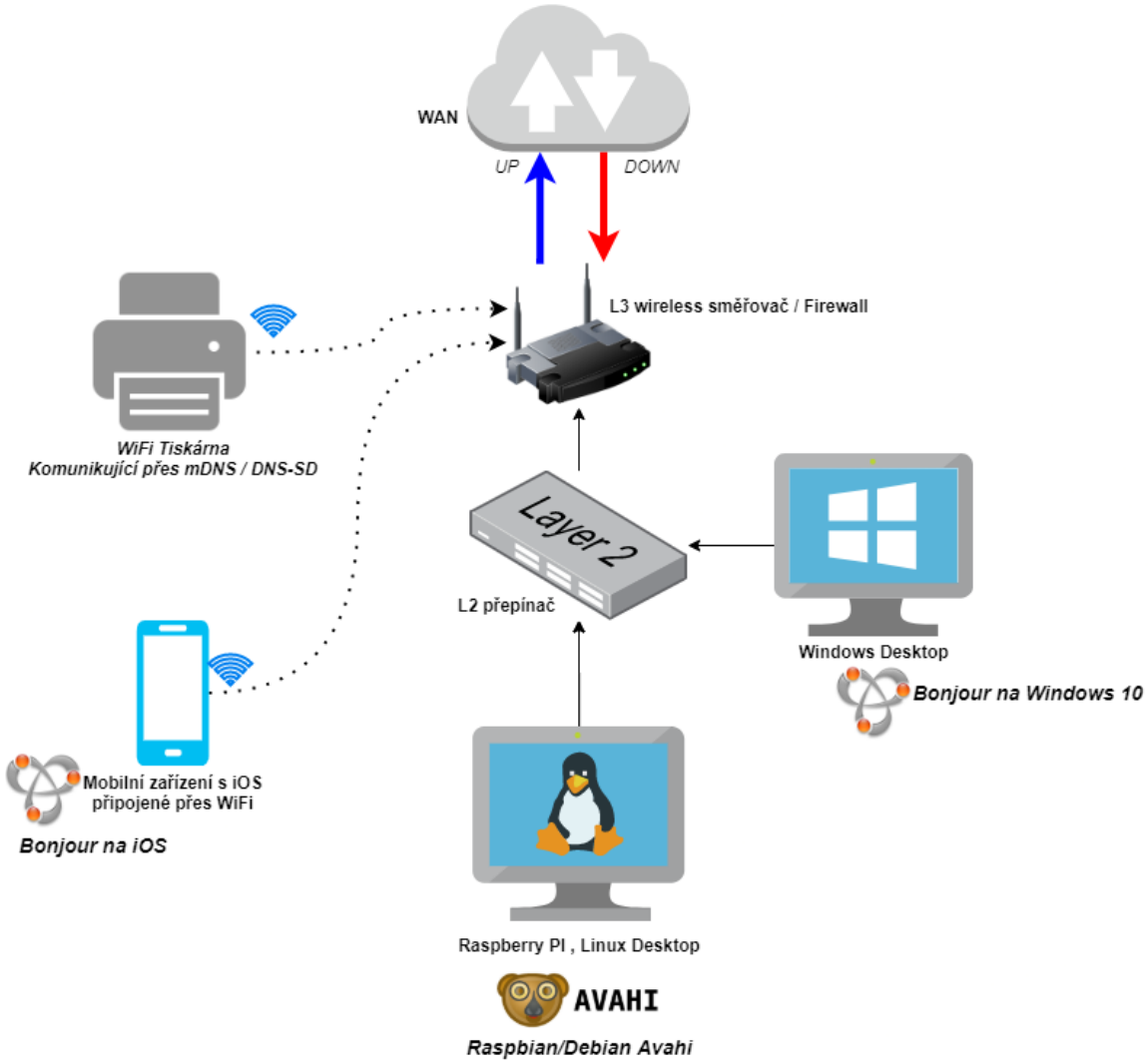
5.1.1 Testovací implementace v domácím prostředí - Avahi / Bonjour

Testovácí implementace v domácím prostředí, ve které byla odzkoušena kooperace služeb Avahi a Bonjour. Jedná se již o zapojenou a využívanou síť. Tato síť obsahuje jeden hlavní počítač s operačním systémem Windows 10, Raspberry PI s operačním systémem Raspbian, mobilní zařízení s operačním systémem iOS, tiskárna pracující s mDNS/DNS-SD službami pro tisk a Xiaomi čistička vzduchu. Viz tabulka zařízení **2**.

V této implementaci bylo odzkoušeno připojení na tiskárnu a čističku vzduchu pomocí služby Avahi a Bonjour, nikoliv sdílení souborů přes službu Samba.

Tabulka 2: Seznam zařízení v domácí implementaci

Seznam zařízení			
Název:	**Operační systém:**	**Avahi / Bonjour:**	**Zeroconf služby:**
Desktop-Lukas.local	Windows 10	Bonjour	/
PIhole.local	Raspbian	Avahi	HTTPs , SSH
Lukas-iPhone.local	iOS	Bonjour	AirPlay, AirDrop
HPC4651672EF27.local	/	obojí	IPP, IPPs, PDL, HTTPs
Xiaomi-airpurifier.local	/	obojí	MIIO
L2 přepínač	/	/	/
L3 směřovač	ubnts	/	/



Obrázek 4: Testovací implementace v domácím prostředí - Avahi / Bonjour

5.1.2 Implementace Avahi ve školní laboratoři

V této implementace služby Avahi jsem se zaměřili na ověření funkčnosti tohoto balíčku s využitím služby Samba pro sdílení souborů skrze místní síť. Všechny instalace probíhali na GNU/-Linux distribucích. Jednalo se o distribuci Ubuntu a Raspbian (Debian based).

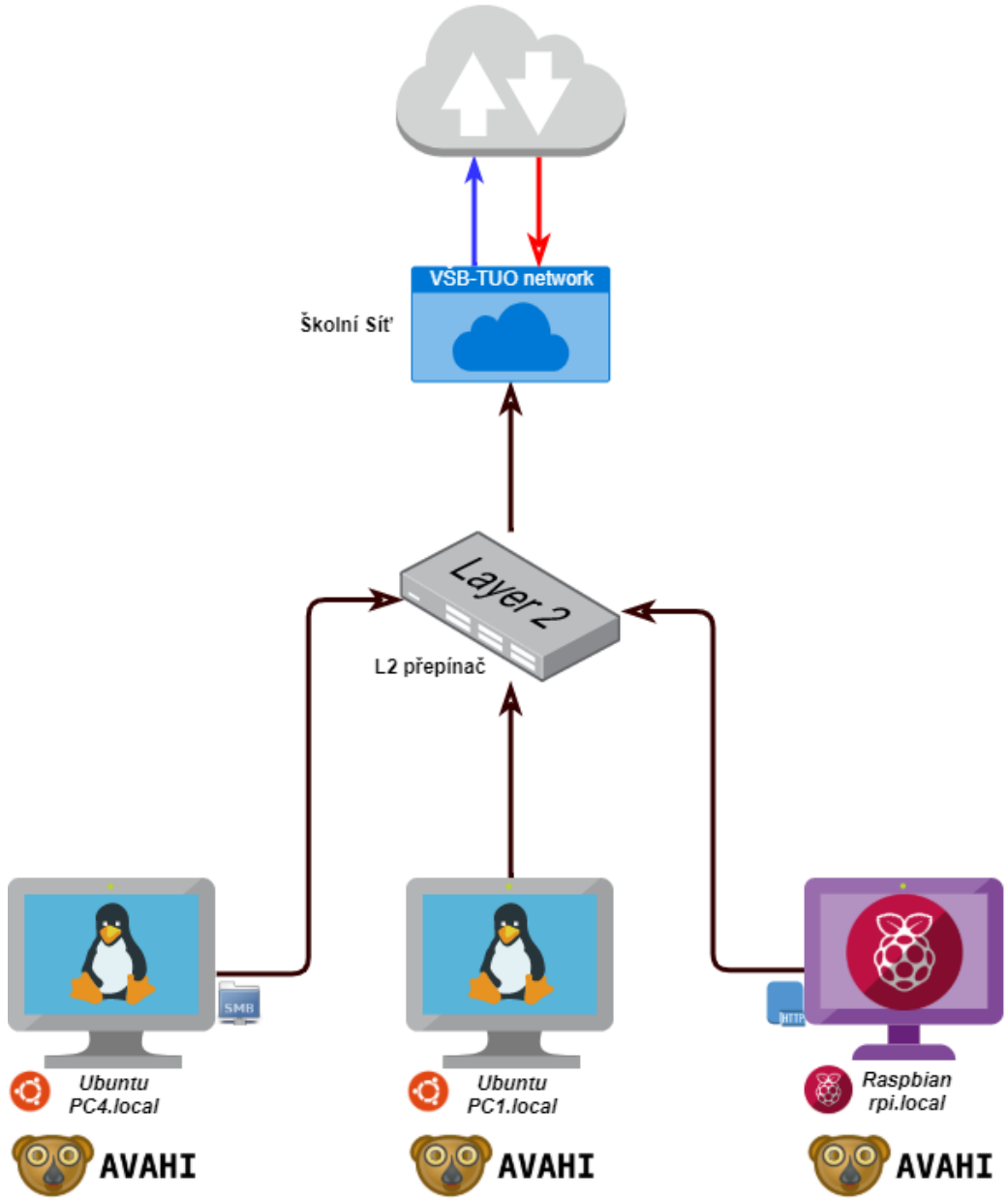
Síťové zapojení bylo navrženo, nebo spíše převzato, ze schématu zapojení školní laboratoře. Pro otestování těchto služeb stačilo pouze pár počítačů a Raspberry PI jako SMB server.

Dle schématu na obrázku **5.1.2** lze vidět zapojení dvou počítačů a raspberry PI do Layer2 ISO/OSI přepínače, který byl poté zapojen do školní sítě, která zprostředkovává WAN konektivitu a další důležité služby, jako DHCP apod. . Toto jednoduché zapojení může být rozšířeno i o další prvek, jako například Layer3 směřovač, který by simuloval reálné síťové zapojení, nicméně je tento prvek nepodstatný a v našem případě se tento prvek nachází někde ve školní síti.

Dle označení v místnosti byly využity počítače s číselným označením 1 a 4. Toto číselné označení následně jsme využili i jako název zařízení a zároveň jako doménové jméno pro Zeroconf local doménu. Seznam zařízení viz tabulka **3**.

Tabulka 3: Seznam zařízení ve školní implementaci

Seznam zařízení			
Název:	**Operační systém:**	**Avahi / Bonjour:**	**Zeroconf služby:**
PC1.local	Ubuntu	Avahi	SSH, SMB
PC4.local	Ubuntu	Avahi	SSH, SMB
Raspberry PI	Raspbian	Avahi	SSH, SMB
L2 přepínač	cisco IOS	/	/



Obrázek 5: Implementace Avahi ve školní laboratoři

5.1.3 Implementace Avahi ve virtuálním prostředí

Implementace ve virtuálním prostředí je poslední implementace, ve které byla otestována funkčnost služby Avahi na GNU/Linux serverových distribucích. V této implementaci jsme využili Ubuntu server GNU/Linux distribuci, která je sice téměř shodná se svým Desktop protějškem, avšak mohou nastat nečekané problémy a jiný způsob implementace Avahi v průběhu procesu.

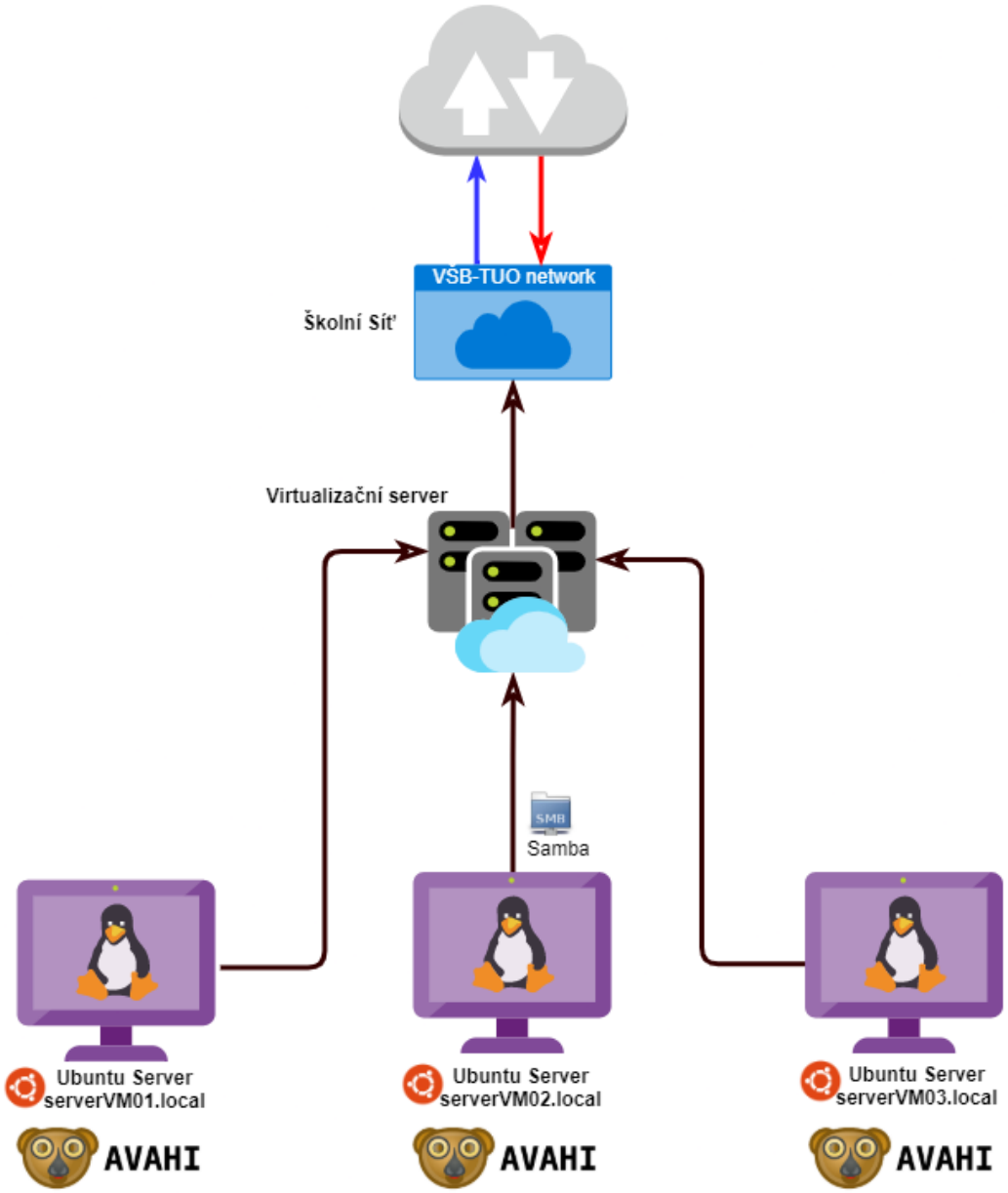
Zapojení, pokud se to tak dá nazvat, bylo provedeno ve virtuálním prostředí na třech stanicích se systémem Ubuntu server. Tyto stanice byly vytvořeny na serveru, který byl zprostředkován vedoucím práce Ing. Pavlem Nevludem. Jedná se o server pro výuku, který je umístěn v serverové místnosti na FEI.

Z pohledu síťového zapojení jsou všechny tyto virtuální instance (servery) propojeny přes virutální síťové adaptéry zprostředkované virtualizačním systémem, který agreguje reálné rozhraní serveru. Tento server je pak připojen do školní sítě. Seznam zařízení viz tabulka **4**.

Testování této implementace spočívalo pouze v práci v konzoli těchto virtuálním serverů, vzhledem k tomu, že GNU/Linux serverové distribuce Ubuntu nemají grafické rozhraní operačního systému, pouze konzoli (CLI).

Tabulka 4: Seznam zařízení ve virtuální implementaci

Seznam zařízení			
Název:	**Operační systém:**	**Avahi / Bonjour:**	**Zeroconf služby:**
serverVM01.local	Ubuntu Server	Avahi	SSH, SMB
serverVM02.local	Ubuntu Server	Avahi	SSH, SMB
serverVM03.local	Ubuntu Server	Avahi	SSH, SMB
Virtualizační server	Ubuntu server	/	/



Obrázek 6: Implementace Avahi ve virtuálním prostředí

5.2 Instalace služby - Avahi

5.2.1 Instalace služby - Ubuntu

1. Nejprve se provedla aktualizaci dostupných repositářů pro instalaci aktuálních balíčků a služeb.

```
sudo apt-get update
```

Výpis 1: Stažení aktualizací repositářů

2. Poté byla provedena instalace dostupných aktualizací pro repositáře

```
sudo apt-get upgrade
```

Výpis 2: Instalace aktualizací repositářů

3. Následně byla instalována hlavní část služby Avahi a to Avahi-Daemon, který nám zajišťuje chod technologie Zeroconf. Jedná se o páteřní službu, bez které se Avahi neobejde.

```
sudo apt-get install avahi-daemon
```

Výpis 3: Instalace Avahi-Daemon

4. Následně jsme mohli doinstalovat dodatečné pod balíčky služby Avahi, díky kterým můžeme poté službu Avahi plně využívat pro prohlížení, sdílení a objevování služeb. Avahi je bez tohoto balíčku funkční, ale přidá nám další parametry do konzole pro manuální odzkoušení služby.

```
sudo apt-get install avahi-utils
```

Výpis 4: Instalace Avahi-Utils

5. Dodatečně jsme mohli doinstalovat i pod balíček Avahi Discover, který nám umožní využívat grafické rozhraní, ve kterém i bez znalostí syntaxe příkazů pro Avahi můžeme objevovat služby.

```
sudo apt-get install avahi-discover
```

Výpis 5: Instalace Avahi-Discover

6. Pokud se nám automaticky s Avahi-Daemon nenainstaloval i Avahi-Autoipd pod služba, může být provedena manuální doinstalace. Ve většině případů se ale tato služba instaluje zároveň s Avahi-Daemonem

```
sudo apt-get install avahi-autoipd
```

Výpis 6: Instalace Avahi-Autoipd

5.2.2 Instalace služby - Raspbian

1. Nejprve jsme provedli aktualizaci dostupných repositářů pro instalaci aktuálních balíčků a služeb.

```
sudo apt-get update
```

Výpis 7: Stažení aktualizací repositářů

2. Poté jsme provedli instalace dostupných aktualizací pro repositáře

```
sudo apt-get upgrade
```

Výpis 8: Instalace aktualizací repositářů

3. Následně následně jsme nainstalovali hlavní část služby Avahi a to Avahi-Daemon, který nám zajišťuje chod technologie Zeroconf. Jedná se o páteřní službu, bez které se Avahi neobejde.

```
sudo apt-get install avahi-daemon
```

Výpis 9: Instalace Avahi-Daemon

4. Následně jsme mohli doinstalovat dodatečné pod balíčky služby Avahi, díky kterým můžeme poté službu Avahi plně využívat pro prohlížení, sdílení a objevování služeb. Avahi je bez tohoto balíčku funkční, ale přidá nám další parametry do konzole pro manuální odzkoušení služby.

```
sudo apt-get install avahi-utils
```

Výpis 10: Instalace Avahi-Utils

5. Dodatečně jsme mohli doinstalovat i pod balíček Avahi Discover, který nám umožní využívat grafické rozhraní, ve kterém jsme i bez znalostí syntaxe příkazů pro Avahi mohli objevovat služby.

```
sudo apt-get install avahi-discover
```

Výpis 11: Instalace Avahi-Discover

6. Pokud se nám automaticky s Avahi-Daemon nenainstaloval i Avahi-Autoipd pod služba, můžeme provést manuální doinstalaci. Ve většině případů se ale tato služba instaluje zároveň s Avahi-Daemonem

```
sudo apt-get install avahi-autoipd
```

Výpis 12: Instalace Avahi-Autoipd

5.2.3 Instalace na jiných GNU/Linux distribucích

V případě instalace Avahi na jiných GNU/Linux distribucích se jména balíčků mohou lišit. Některé distribuce, jak už bylo zmíněno, mají již tuto službu předem předinstalovanou, nicméně se může stát (hlavně v případě CLI-only systémů / serverů), že tato služba bude chybět.

Fedora

GNU/Linus distribuce Fedora je jedna z nejpoužívanějších komunitních distribucí GNU/Linux. V této distribuci se většinou Avahi balíček od základu nenachází a je třeba ho doinstalovat. Instalaci můžeme provést pomocí těchto příkazů:

```
sudo dnf install avahi
sudo dnf install avahi-autoipd
sudo dnf install avahi-compat-libdns_sd
sudo dnf install avahi-glib
sudo dnf install avahi-gobject
sudo dnf install avahi-tools
sudo dnf install nss-mdns
```

Výpis 13: Instalace Avahi

ArchLinux

PACMAN based komunitní GNU/Linux distribuce, ve které se ve výchozím nastavení také nenachází balíček Avahi. Lze jej jednoduše doinstalovat pomocí příkazu:

```
sudo pacman -S avahi
```

Výpis 14: Instalace Avahi

5.2.4 Konfigurace Avahi Daemona

V případě, že nechceme využívat základní parametry služby Avahi, lze dodatečné parametry nakonfigurovat v konfiguračním souboru avahu-daemon.conf . V tomto souboru můžeme změnit parametry jako doménové jméno, doménu, dodatečné domény, využití IPv4 a IPv6 a další.

Popis nejdůležitějších parametrů v avahi-daemon.conf :

- pomocí parametru host-name jsme nastavili stanici jméno, které bude využívat v základní lokální doméně ".local". Dalšími dvěmi parametry můžeme změnit základní doménu pro Zeroconf, nebo případně přidat další domény, které do tohoto procesu budou také spadat. Nicméně změna základní domény není silně doporučována, můžu způsobit chybné překlady, nebo případně spory v DNS dotazech na síti,

```
[server]
#host-name=foo
#domain-name=local
#browse-domains=0pointer.de, zeroconf.org
```

- těmito dalšími užitečnými parametry můžeme službě Avahi specifikovat, jaké verze internetového protokolu má využívat. V našem případě implementace ve školní laboratoři byl využit pouze parametr pro IPv6,

```
use-ipv4=yes
use-ipv6=yes
```

- lze zvolit, na kterých rozhraních bude služba Avahi aktivní,

```
#allow-interfaces=eth0
#deny-interfaces=eth1
```

- pokud bychom chtěli omezit i maximální počet uživatelů a skupin, které budou využívat službu Avahi, můžeme toto nastavit pomocí těchto parametrů,

```
#clients-max=4096
#objects-per-client-max=1024
#entries-per-entry-group-max=32
```

- v konfigurační části označené jako "publish"lze nastavit parametry pro sdílení služeb. Zejména lze povolit, nebo zakázat sdílení na celé stanici, nebo pouze uživatelům dané stanice,

```
[publish]
#disable-publishing=no
#disable-user-service-publishing=no
```

- tyto parametry jsou v základu zakázány, avšak mohou být velmi užitečné pro prvotní nastavení a pochopení sdílení služeb přes Avahi. Tyto parametry umožňují sdílet informace o stanici,

```
publish-hinfo=no
publish-workstation=no
```

- pokud budeme chtít sdílet doménu nebo případně informace o našich nastavených DNS serverech, lze využít tyto, v základu nepoužité, parametry. Případně můžeme povolit zápis mDNS responderů do resolv.conf konfiguračního souboru,

```
#publish-domain=yes
#publish-dns-servers=192.168.50.1, 192.168.50.2
#publish-resolv-conf-dns-servers=yes
```

5.2.5 Konfigurace služeb

Všechny služby jsou uloženy jako XML fragmenty, které obsahují statické DNS-SD data. Každý soubor může obsahovat více definic služeb, které sdílejí stejný název. To je užitečné pro publikování dat pro služby, které implementují více protokolů. (tj. tiskárna implementující _ipp._tcp a _printer._tcp)

```
<!DOCTYPE service-group SYSTEM "avahi-service.dtd">
<service-group>
```

Parametr service-group by měl obsahovat jeden nebo více parametrů (name / service).

```
<name>[RPI] SMB file sharing</name>
```

Pokud budeme chtít specifikovat parametr "service", je nutné přidat dodatečné parametry pro jeho typ a port. Dodatečně lze přidat "host-name", "domain-name"a "subtype"parametry.

```
<service>
<type>_smb._tcp</type>
<port>139</port>
</service>
</service-group>
```

5.2.6 Volitelná konfigurace souboru hosts

Tento konfigurační soubor je téměř totožný s konfiguračním souborem uložených v "etc/hosts", ve kterém lze manuálně definovat A a AAAA DNS záznamy, nebo manuálně zapsat IP adresace ke konkrétním doménovým jménům. Celý výpis tohoto souboru můžete nalézt v příloze **3.4.6**

Manuální zápis v konfiguračním souboru hosts musíme zapsat ve formátu IP adresa a název stanice.

```
127.0.0.1 rpi
```

5.2.7 Opravná konfigurace nsswitch (mDNS)

Pokud budeme chtít využívat službu Avahi v GNU/Linux prostředí, s největší pravděpodobností budeme muset upravit soubor nsswitch.conf uložený v adresáři etc. Jedná se o nss-mdns lib balíček, který zajišťuje překlad pomocí mDNS. Tento balíček byl naposledy aktualizován v roce 2018 a v několika GNU/Linux distribucích může jeho základní konfigurace způsobovat problémy s překladem. Minimálně se jedná o distribuce Ubuntu,Fedora a ArchLinux, jsou kde tyto problémy nahlášeny uživateli.
Zápis parametru v aktuální verzi tohoto balíčku není přesně specifikován , nicméně, po hlubším průzkumu diskuzí a ověřování různých konfigurací jsme se dobrali k tomuto optimálnímu parametru.

```
hosts:  files mdns4_minimal mdns6_minimal [NOTFOUND=return] mdns4 mdns6 dns
```

5.2.8 Spuštění a vypnutí služby Avahi

Jedním ze způsobů, jak můžeme inicializovat (případně restartovat a zastavit) službu Avahi v Ubuntu a Debian GNU/Linux distribucích je, že manuálně spustíme skript Avahi Daemona pomocí těchto příkazů:

```
sudo /etc/init.d/avahi-daemon start

sudo /etc/init.d/avahi-daemon restart

sudo /etc/init.d/avahi-daemon stop
```

Dalším způsobem pro správu Avahi Daemon procesu je využití systemctl a systemd příkazů.

```
sudo systemctl start avahi-daemon.conf

sudo systemctl restart avahi-daemon.conf

sudo systemctl stop avahi-daemon.conf
```

Další parametry:

Dodatečné parametry pro práci s Avahi-Daemonem jsou doplněny v tabulce **5**.

Zkrácený přepínač:	Přepínač:	Popis:
-f	*–file*	Tímto parametrem můžeme specifikovat konfigurační soubor, který nahradí základní avahi-daemon.conf .
-D	*–daemonize*	Zapnutí služby Avahi (avahi-daemon) po zapnutí systému.
-s	*–syslog*	Nastavení logování do syslogu, narozdíl od STDERR.
-k	*–kill*	Zastavení procesu Avahi-Daemon (ekvivalent pro příkaz SIGTERM).
-r	*–reload*	Řekněte již spuštěnému avahi-daemonovi, aby si znovu přečetl /etc/resolv.conf (v případě, že jste ve avahi-daemon.conf povolili publikovat resolv-conf-dns-servery v avahi-daemon.conf) soubory z /etc/avahi/services/. Vezměte prosím na vědomí, že tím nebude znovu načten soubor /etc/avahi/avahi-daemon.conf. (ekvivalent k odeslání SIGHUP).
-c	*–check*	Navrátí hodnotu "0", pokud je již proces spuštěn.
-h	*–help*	Zobrazení nápovědy.
-v	*–version*	Zobrazení verze Avahi.
/	–debug	Zobrazí podrobnosti při logování.
/	–no-rlimits	(popis není dodán v manuálu)
/	–no-chroot	(popis není dodán v manuálu)
/	–no-proc-title	Není popsán v manuálu, pouze je dodána poznámka, aby se neměnilo jméno procesu v případě, že běží.

Tabulka 5: Tabulka parametrů pro Avahi-Daemon

5.2.9 Vyhledání služeb pomocí Avahi-Browse

Pokud bychom chtěli vyhledávat služby na místní síti (LAN), můžeme využít službu Avahi-Browse. Pokud bychom chtěli vyhledat všechny služby, můžeme využít tento příkaz:

```
avahi-browse --all
```

Další parametry:

Dodatečné parametry pro práci s Avahi-Browse jsou doplněny v tabulce **6**.

Zkrácený přepínač:	**Přepínač:**	**Popis:**
-a	*-all*	Hledání všech dostupných služeb na místní síti.
-D	*–browse-domains*	Hledání všech dostupných domén, místo služeb, na místní síti.
-d	*–domain={DOMAIN}*	Hledání služeb ve specifické doméně.
-v	*–verbose*	Zapnutí "verbose"režimu.
-t	*–terminate*	Ukončení procesu po uložení seznamu služeb.
-c	*–cache*	Ukončení procesu po vymazání všech záznamů.
-I	*–ignore-local*	Ignoruje služby na lokální stanici.
-r	–resolve	Automaticky překládá detaily u nalezených služeb.
-f	–no-fail	Podmínka pro neukončení procesu, i když hledání nic nenajde.
-p	–parsable	Zjednodušší výstup pro pozdější analýzu, nebo použití ve skriptech. Parametr rozděluje služby pomocí ";". Doporučuje se využití s parametrem –no-db-lookup .
-k	–no-db-lookup	Nevyhledává služby v databázi.
-b	–dump-db	Vymaže databázi, dá se kombinovat s přepínačem –kill.
-h	–help	Zobrazení nápovědy.
-V	–version	Zobrazení verze Avahi.

Tabulka 6: Tabulka parametrů pro Avahi-Browse

5.2.10 Generování IPv4 LL adresace

Pokud bychom chtěli na rozhraní přidělit IPv4 LL adresu, lze využít tento příkaz, kde použijeme přepínač start a určíme název rozhraní, pro které se má adresa generovat.

```
avahi-autoipd --start nazev_rozhrani
```

Další parametry:

Dodatečné parametry pro práci s Avahi-Autoipd jsou doplněny v tabulce **7**.

Zkrácený přepínač:	Přepínač:	Popis:
-D	–*daemonize*	StartFragmentZapnutí služby Avahi (avahi-autoipd) po zapnutí systému. EndFragment
-w	–*wait*	Čeká dokud není vygenerována / přidělena adresa. Validná pouze s přepínačem –daemonize .
-s	–*syslog*	Nastavení logování do syslogu, narozdíl od STDERR.
-k	–*kill*	Zastavení procesu Avahi-Daemon (ekvivalent pro příkaz SIGTERM).
-S	–*start={IP}*	Pokusí se přidělit specifikovanou IPv4 adresu. Adresa musí být z IPv4 LL subnetu 169.254.0.0/16 .
-h	–*help*	Zobrazení nápovědy.
-v	–*version*	Zobrazení verze Avahi.
/	–debug	Zobrazí podrobnosti při logování.
/	–force-bind	Vynutí nastavení adresy i v případě, že již platná adresa byla nastavena na rozhraní.
/	–no-chroot	(popis není dodán v manuálu)
/	–no-proc-title	Není popsán v manuálu, pouze je dodána poznámka, aby se neměnilo jméno procesu v případě, že běží.

Tabulka 7: Tabulka parametrů pro Avahi-Autoipd

Pokud neznáme název rozhraní, můžeme si vyjet síťovou konfiguraci pomocí:

```
:~ $ ifconfig

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.100.254 netmask 255.255.255.0 broadcast 192.168.100.255
        inet6 fe80::679a:374f:2a7:1eec prefixlen 64 scopeid 0x20<link>
        inet6 2a00:ca8:a1f:ea5b::1003 prefixlen 128 scopeid 0x0<global>
        ether b8:27:eb:47:e5:89 txqueuelen 1000 (Ethernet)
        RX packets 17525040 bytes 1126094387 (1.0 GiB)
        RX errors 24024 dropped 0 overruns 0 frame 0
        TX packets 28658004 bytes 1567153232 (1.4 GiB)
        TX errors 177 dropped 0 overruns 0 carrier 0 collisions 0
```

nebo

```
:~ $ ip a

2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
    group default qlen 1000
    link/ether b8:27:eb:47:e5:89 brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.254/24 brd 192.168.100.255 scope global noprefixroute eth0
       valid_lft forever preferred_lft forever
    inet6 2a00:ca8:a1f:ea5b::1003/128 scope global dynamic noprefixroute
       valid_lft 58218sec preferred_lft 58218sec
    inet6 fe80::679a:374f:2a7:1eec/64 scope link
       valid_lft forever preferred_lft forever
```

5.2.11 Vyhledání služeb pomocí Avahi-Discover

Pokud uživatel nemá praxi v práci s příkazovým řádkem v prostředí GNU/Linux, je možné doinstalovat balíček Avahi-Discover, který nahrazuje službu Avahi-Browse pro vyhledávání služeb na místní síti (LAN).

Abychom mohli využít grafické rozhraní, musí mít operační systém GUI (nesmí to být připojení na CLI, nebo CLI-only systém). Poté lze v CLI spustit Avahi-Discover pomocí tohoto příkazu:

```
avahi-discover
```

pokud bychom se pokusili vyvolat tuto akci přes SSH, nebo v CLI-only systému, dostaneme tuto chybu:

```
~ $ avahi-discover
Unable to init server: Could not connect: Connection refused
Unable to init server: Could not connect: Connection refused

(avahi-discover:9784): Gtk-CRITICAL **: 00:42:13.192:
    _gtk_style_provider_private_get_settings: assertion '
    GTK_IS_STYLE_PROVIDER_PRIVATE (provider)' failed

(avahi-discover:9784): Gtk-CRITICAL **: 00:42:13.192:
    _gtk_style_provider_private_get_settings: assertion '
    GTK_IS_STYLE_PROVIDER_PRIVATE (provider)' failed

(avahi-discover:9784): Gtk-CRITICAL **: 00:42:13.192:
    _gtk_style_provider_private_get_settings: assertion '
    GTK_IS_STYLE_PROVIDER_PRIVATE (provider)' failed
```

5.3 Instalace služby - Samba / CIFS

5.3.1 Instalace služby Samba

```
sudo apt-get install samba
```

5.3.2 Instalace služby CIFS

```
sudo apt-get install cifs-utils
```

5.3.3 Popis konfiguračního souboru pro SMB

Pokud chceme využívat správně službu SMB/CIFS, musíme nejprve pochopit, jak správně nastavit konfigurační soubor pro tuto službu. Většina parametrů je detailně popsána již v tomto konfiguračním souboru a také v manuálu SMB. Výpis celého konfiguračního souboru nalezneme v příloze **B.9**

Popis konfiguračního souboru SMB.conf:

- jako první parametry v tomto souboru lze nalézt parametry ve skupině "global", jedná se o globální, nebo základní parametry této služby. Tato hlavička je zapsána takto:

```
#======================= Global Settings =======================
[global]
```

- prvními parametry, které mohou být upraveny v tomto konfiguračním souboru, jsou parametry určující doménu, nebo pracovní skupinu, popisek pro službu SMB, WINS server a nastavení NetBIOS DNS proxy,

```
workgroup = WORKGROUP
server string = %h server (Samba, Ubuntu)
wins server = w.x.y.z
dns proxy = no
```

- další parametry jsou navázány na síťové nastavení. Prvním příkazem můžeme určit v jakém subnetu má SMB pracovat, případně název rozhraní. Druhým parametrem se lze vázat pouze na pojmenovaná rozhraní (např. eth0). Je doporučeno nechat tento parametr povolený,

```
interfaces = 127.0.0.0/8 eth0
bind interfaces only = yes
```

- můžeme také nastavit způsob logování, cestu pro logovaní soubor, jeho maximální velikost a parametry spojené se syslogem. Případně můžeme nastavit "panic action", který je užitečný v případě, že by služba spadla (můžeme například odeslat email administrátorovi),

```
log file = /var/log/samba/log.%m
max log size = 1000
syslog only = no
syslog = 0
panic action = /usr/share/samba/panic-action %d
```

- v případě, že budeme chtít řešit podrobněji autentizaci, můžeme vyeditovat tyto parametry:

```
server role = standalone server
passdb backend = tdbsam
obey pam restrictions = yes
passwd program = /usr/bin/passwd %upasswd chat = *Enter\snew\s*\spassword
    :* %n\n *Retype\snew\s*\spassword:* %n\n *password\supdated\
    ssuccessfully* .
pam password change = yes
map to guest = bad user
```

- tato sekce parametrů definuje nastavení pro využití SMB jako Active Directory server,

```
logon path = \\%N\profiles\%U
logon drive = H:
logon script = logon.cmd
add user script = /usr/sbin/adduser --quiet --disabled-password --gecos ""
     %u
add machine script = /usr/sbin/useradd -g machines -c "%u machine account"
     -d/var/lib/samba -s /bin/false %u
add group script = /usr/sbin/addgroup --force-badname %g
```

- poslední smíšené a nezařazené parametry globálního nastavení SMB,

```
;   include = /home/samba/etc/smb.conf.%m
;   idmap uid = 10000-20000
;   idmap gid = 10000-20000
;   template shell = /bin/bash
;   usershare max shares = 100
```

- toto jsou sdílené definice služby SMB, můžeme tam najít nastavení pro připojení k osobním složkám uživatelů (a jejich práva), využití NetLogon pro autentizaci uživatelů z SMB Active Directory a nastavení sdílených tiskáren skrze službu SMB. Pod tyto sdílené definice je také vhodné dávat vlastní definice pro připojení pomocí SMB, jako například práva pro přístup ke všem souborům pro konkrétní uživatele (administrátory) a podobně. Viz příloha **B.8**.

5.3.4 Konfigurace SMB

Pokud bychom chtěli vytvořit vlastní sdílenou definici pro připojení a sdílení souborů pomocí služby SMB, musíme tyto parametry doplnit do konfiguračního souboru SMB.conf, který se nachází zde:

```
/etc/samba/smb.conf
```

Na libovolné místo lze za globální parametry vložit naši vlastní definici. Nicméně je doporučeno vkládat vlastní definice na konec dokumentu. Příklad vlastní definice, která je také použita v implementaci Zeroconf této práce:

```
[Zeroconf Filesharing]
        path = /etc/
        writeable = no
        browseable = yes
        read only = no
        write list = student, pi
        read list = student, pi
        valid users = student, pi
        security = user
```

5.4 Instalace služby - Bonjour

Služba Bonjour je jednou z dalších možností, jak na zařízení implementovat technologii Zeroconf. V této práci jsme se věnovali instalaci v prostředí operačních systémů Windows, MAC OS a iOS vzhledem k tomu, že tyto systémy jsou oficiálně podporovány společností Apple pro instalaci této služby.

Podpora v GNU/Linux prostředí je teoreticky možná, pouze pokud uživatelé vymysleli port této implementace, nicméně společnost Apple se ve svých dokumentacích o ní nezmiňuje, proto instalaci v operačních systému GNU/Linux jsme vynechali.

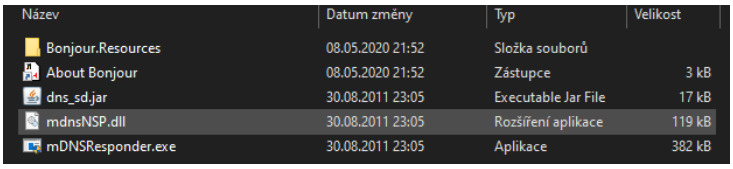
5.4.1 Instalace služby - Windows

V prostředí operačního systému Windows (zejména Windows 7 a výše) je implementace Bonjour možná pomocí instalačního souboru, který má Apple dostupný ke stažení na svých Developer stránkách nebo se instalace provede automaticky, pokud do systému nainstalujeme aplikaci iTunes.

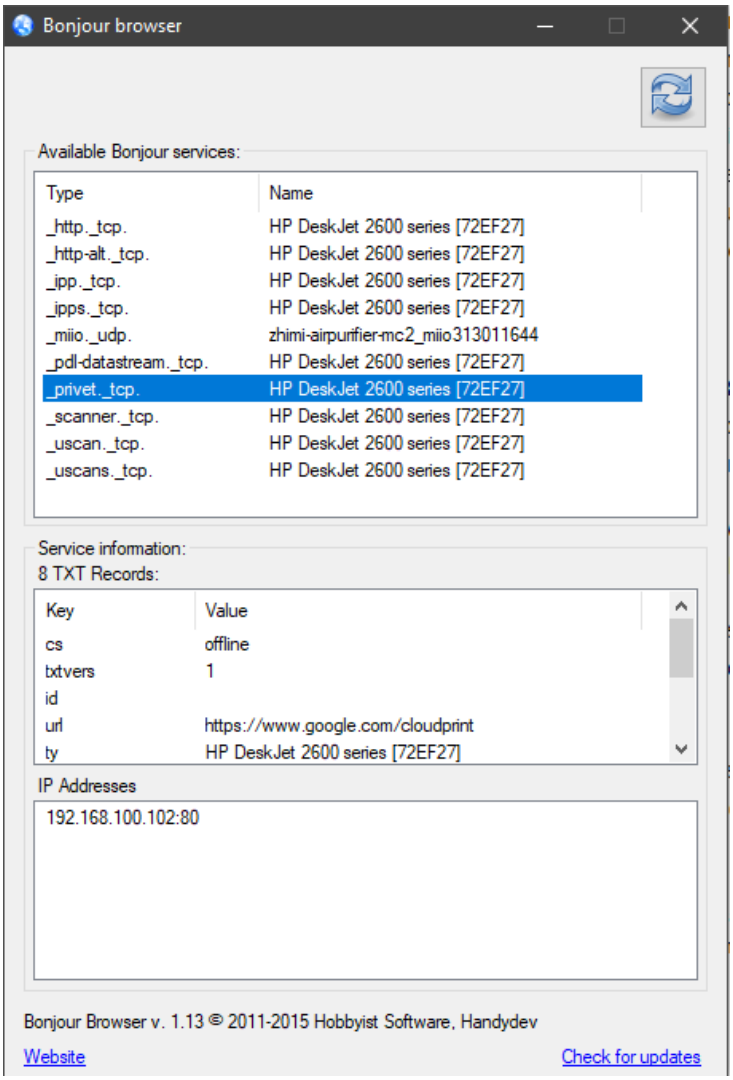
Instalace samotné služby Bonjour, v tomto OS, je jednoduchá a intuitivní, jako většina instalací. Stačí pouze spustit instalační soubor, proklikat se kroky "dále"a můžeme využívat technologii Zeroconf.

Na stránce **"developer.apple.com"** jsou na výběr dvě varianty služby Bonjour, jednou z nich je BonjourPS určená pouze pro mDNS komunikaci s tiskárnami a druhou variantou je BonjourSDK, určena pro klasickou instalaci. Pro stažení BonjourSDK je nutné se přihlásit, nebo si vytvořit AppleID.

Všechny soubory této Zeroconf implementace jsou uloženy dle cesty, kterou uvádíme při instalaci. Ověření funkčnosti můžeme provést pomocí dodatečné aplikace pro Windows **Bonjour-Browser**, která je zdarma ke stažení například ze stránek **"www.hobbyistsoftware.com"**. Viz obrázky **7a** a **7b**



(a) Soubory služby Bonjour uložené na stanici s OS Windows 10



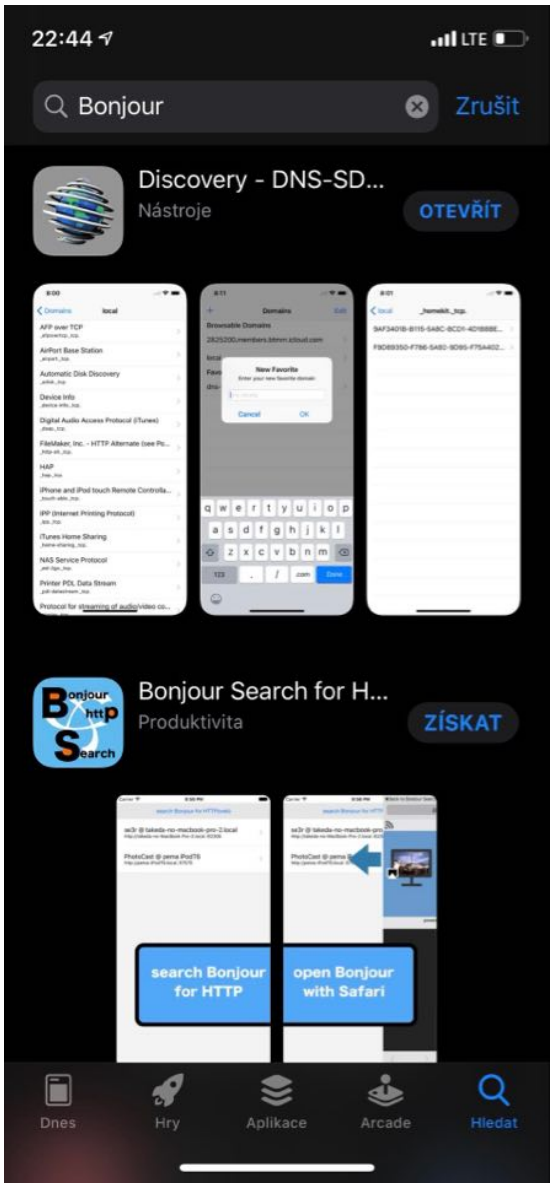
(b) Ověření funkčnosti mDNS pomocí aplikace BonjourBrowser

Obrázek 7: Ověření funkčnosti technologie Zeroconf pomocí služby Bonjour

5.4.2 Instalace služby - macOS / iOS

Instalace na těchto OS není nutná vzhledem k tomu, že firma Apple do svých OS tuto službu implementuje automaticky a využívá ji pro své další služby, jako například AirPlay nebo AirDrop (aj.).

Pro ověření funkčnosti můžeme nainstalovat BonjourBrowser, nebo DNS-SD Browser z AppleStoru. Viz obrázek **8**



Obrázek 8: Aplikace pro ověření funkčnosti služby Bonjour (mDNS a DNS-SD) na MACOS a iOS

5.5 Chyby a nedostatky technologie Zeroconf

5.5.1 Avahi a Bonjour chyba s VPN

Samotná společnost Apple přiznala nedostatky v aktuální verzi Bonjour, problémy se mohou vyskytnout při používání VPN a Zeroconf technologie. Mohou nastat výpadky v prohledávání a odpovědích v mDNS, nebo DNS-SD procesu. Dá se očekávat, že stejné problémy budou i u služby Avahi, která je velmi podobná (téměř totožná) službě Bonjour.

5.5.2 Avahi nss-mdns-0.14.1-1 chyba překladu pro IPv4

V aktuální verzi Avahi se mohou objevit problémy s mDNS službou, problémy jsou zejména hlášeny u GNU/Linux distribucí Ubuntu, Fedora a ArchLinux. Problém nastává v konfiguračním souboru nsswitch.conf , kde jsou špatně nastaveny parametry pro mDNS respondér, doporučené parametry jsou popsány v kapitole **5.2.7**. Tento problém může afektovat komunikaci pomocí IPv6 protokolu.

5.5.3 Avahi nss-mdns-0.14.1-1 chyba překladu pro IPv6

Jak bylo popsáno v předchozí kapitole **5.5.2**, ve službě Avahi můžeme narazit na problém s mDNS respondérem, tento problém je zřejmý, pokud chceme využívat pouze IPv6 adresaci. Služba Avahi se pak chová tak, že sice dokáže objevit nějaké služby, ale již není schopná s němi komunikovat, dokud se neupraví parametry tohoto konfiguračního souboru.

Příklad chyby můžeme nalézt v přiloženém výpisu kódu **5.5.3**

```
student@pc4:~$ ping rpi.local
ping: rpi.local: Name or service not known
```

5.5.4 Avahi publikování IPv6 adres

Avahi má již dlouhé léta nahlášen problém, při kterém nejde navázat spojeníve chvíli, kdy jedna ze stanic využívá IPv6 LL adresaci a druhá pouze IPv6 globální adresu.[48]

Dalším problémem je nepublikování IPv6 LL adresace, pokud na stejném rozhraní již existuje IPv6 globální adresa. Tato chyba se objevila již v roce 2013 a stále není opravena.[49]

Jedním z dalších problémů může být pomalá indikace služby Avahi, při změně IPv6 adresace, byly nahlášeny problémy, které popisují, že Avahi nedokázalo dost rychle tuto změnu publikovat.

Tento problém je hlavně zřejmý, pokud využíváme například technologii 6to4 tunelování, kde IPv6 adresa vzniká na základě IPv4 adresace, která může být dynamicky přidělována v určitých intervalech.[50]

5.5.5 Bonjour chybné zastavení procesu

Ve službě Bonjour může občas dojít k chybě při zastavení procesu, kdy stanice, která vyhledává služby na síti nemusí obdržet "Goodbye"zprávu od jiné stanice, která ruší jednou ze svých služeb. Tato chyba se může objevit, pokud na stanici, kde ukončujeme vysílání, použijeme tento kód:

```
this.logger.info('Stopping P2P mDNS annoucement...');
this.service.stop();
this.bonjour.unpublishAll();
this.bonjour.destroy();
```

5.5.6 Zeroconf a multicastové bouře

Ve velkých LAN sítích může docházet k velkým bouřím (často označované jako broadcastové bouře, i když používají multicast), protože každé zařízení vysílá na základně mDNS / DNS-SD. Tento problém nám může zapřičínit i výpadek sítě, proto se ve větších sítích nedoporučuje Zeroconf technologii nasazovat.

Také můžeme narazit na problém se stejnými jmény v síti, kdy vzniká dohadovací fáze Zeroconfu a ta, v případě více shodných jmen, může způsobit také bouři. Případem může být virtualizační server z implementace **5.1.3**, na kterém jsou všechny virtuální servery hostovány se stejným jménem a mohou tak vyvolat mDNS bouři.

5.5.7 Bezpečnostní nedostatky

DNS Spoofing

Je jedním z hlavních bezpečnostních rizik technologie Zeroconf, v tomto typu útoku využívá útočník neznalost napadeného uživatele, případně využívá jeho nepozornosti a chyb v prohlížení stránek. Cílem útočníka je například při výpadku WAN konektivity, nahradit věrohodné webové zdroje za své falešné. Případně se může pomocí ".local"domény snažit maskovat jako věrohodný zdroj z internetu a využít nepozornosti napadeného uživatele při prohlížení stránek. V praxi to může znamenat, že uživatel do prohlížeče píše dotaz na web ve formátu například **"facebook"** místo **"facebook.com"**, což může způsobit, že technologie Zeroconf doplní tento základní název o svou lokální doménu **".local"** a zobrazit tak napadenému uživateli falešný obsah webových

stránek.

Také se tento typ útoku využívá pro podvody na lokální síti, kde se útočník snaží maskovat cizím názvem stanice a přesvědčit tak uživatele, aby procházel služby přes něj.

Pro zamezení těmto útokům bychom měli využívat ochranných a zabezpečovacích služeb, jako například **DNSSEC** (DNS Security) nebo **SPYC** (Speaking out Your Certificate).

Man in the Middle Attack

U technologie Zeroconf můžeme narazit na jeden z nejpoužívanějších typů útoků, a to na útok Man In the Middle. Během tohoto útoku se útočník snaží tvářit jako dotazované zařízení v síti, nechat přes sebe proudit všechen provoz, analyzovat ho a využít ve svůj prospěch. [51]

Příklad útoku můžeme nalézt na obrázku **9**

.

Obrázek 9: Příklad útoku Man in the Middle Attack

5.6 Analýza provozu

5.6.1 Analýza provozu a zhodnocení implementace v domácím prostředí

Popis postupu implementace

1. Instalace Avahi služeb.

```
student@server:~$ sudo apt-get install avahi-daemon avahi-auotipd avahi-
    dnsconfd avahi-ui-utils avahi-utils
```

2. Dodatečné nastavení konfigurace Avahi Daemonu.

```
[server]
host-name=PIhole
domain-name=local
use-ipv4=yes
use-ipv6=yes
```

3. Vytvoření konfiguračního souboru, který popisuje sdílené služby přes Avahi.

```
<?xml version="1.0" standalone='no'?><!--*-nxml-*-->
<!DOCTYPE service-group SYSTEM "avahi-service.dtd">
<service-group>
  <name replace-wildcards="yes">RPi (%h) over Avahi </name>
  <service>
    <type>_ssh._tcp</type>
    <port>22</port>
  </service>

    <service>
    <type>_smb._tcp</type>
        <port>139</port>
        <port>445</port>
  </service>

</service-group>
```

4. Oprava nsswitch konfiguračního souboru.

```
hosts:          files mdns4_minimal mdns6_minimal [NOTFOUND=return] dns
```

5. Stažení balíčků pro Sambu a CIFS.

```
kashek@PIhole:/home $ sudo apt-get install samba cifs-utils
```

6. Vytvoření složky pro SMB sdílení, vytvoření uživatele, nastavení SMB hesla pro uživatele a přiřazení práv pro sambashare skupinu.

```
kashek@PIhole:/home $ sudo mkdir discord
kashek@PIhole:/home $ sudo chown discord discord
kashek@PIhole:/home $ sudo chmod 2770 discord
kashek@PIhole:/home $ sudo smbpasswd -a discord
kashek@PIhole:/home $ sudo chown kashek:sambashare kashek
kashek@PIhole:/home $ sudo chown discord:sambashare discord
```

7. Přidání vlastní konfigurační skupiny do SMB.conf, která povoluje uživatelům discord a kashek přístup k určité cestě na stanici.

```
[Zeroconf Filesharing]
path = /home/kashek/
writeable = no
browseable = yes
read only = no
write list = discord, kashek
read list = discord, kashek
valid users = discord, kashek
security = user
```

8. Povolení síťových pravidel.

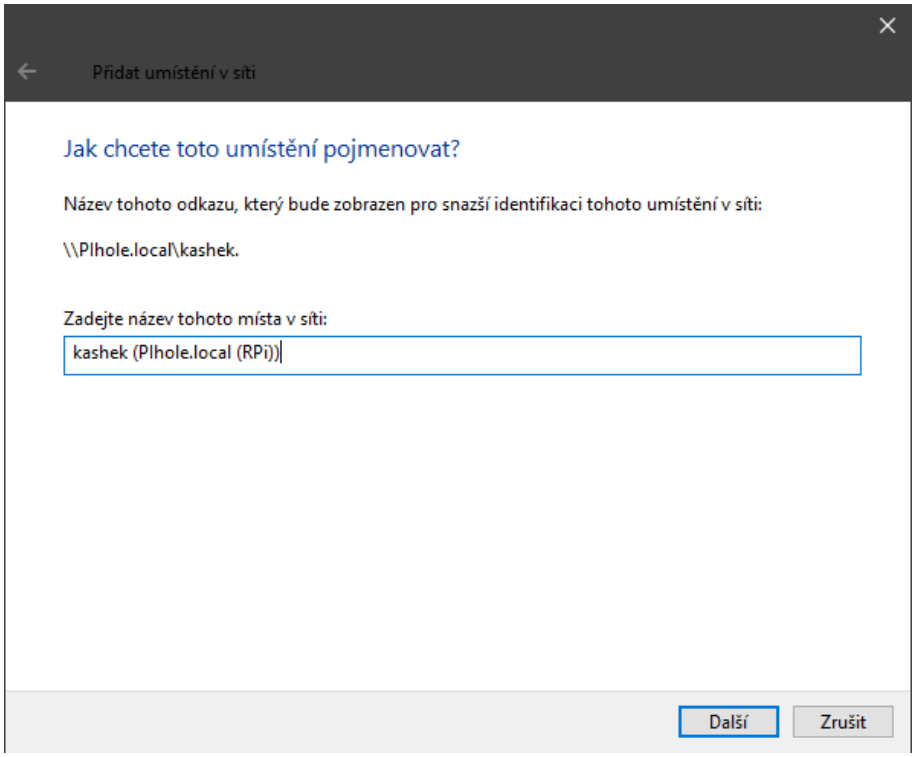
```
kashek@localhost:/etc/samba $ sudo ufw allow samba
Rule added
Rule added (v6)

kashek@localhost:/etc/samba $ sudo ufw allow CIFS
Rule added
Rule added (v6)
```

Ověření funkčnosti

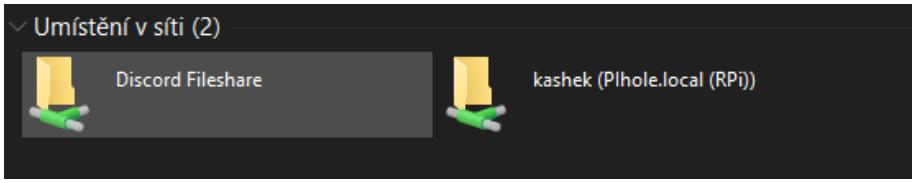
Ověření funkčnosti Zeroconfu pomocí pingu z Microsoft Windows stanice na Raspberry PI GNU-/Linux stanici.

```
C:\Users\major>ping PIhole.local

Pinging PIhole.local [192.168.100.254] with 32 bytes of data:
Reply from 192.168.100.254: bytes=32 time<1ms TTL=64
Reply from 192.168.100.254: bytes=32 time<1ms TTL=64
Reply from 192.168.100.254: bytes=32 time<1ms TTL=64
Reply from 192.168.100.254: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.100.254:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Ověření funkčnosti připojením na SMB server pomocí Zeroconfu. Viz obrázky **10** a **11**



Obrázek 10: Připojení sdílené SMB složky na Microsoft Windows stanici pomocí Zeroconf.

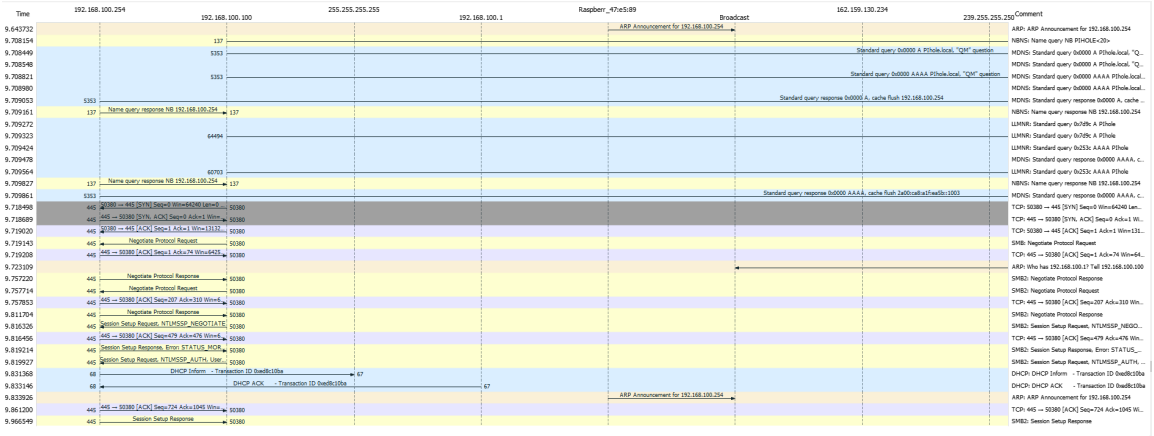


Obrázek 11: Úspěšně připojené síťové složky na Microsoft Windows stanici.

Analýza provozu

Analýza provozu proběhla pomocí programu Wireshark v prostředí operačního systému Microsoft Windows. Na obrázku **12** můžeme vidět úspešně vyhledání služby SMB z Microsoft Windows stanice, připojení pomocí Zeroconf a navázání SMB spojení s úspěšným přenosem souborů.

Na druhém obrázku **13** můžeme vidět UDP multicastový přenos, jeho statistiky a rychlosti.



Obrázek 12: Graf zobrazující odchycenou komunikaci při navázání SMB konexe pomocí mDNS

Wireshark · UDP Multicast Streams · homelab.pcap

Source Address	Source Port	Destination Address	Destination Port	Packets	Packets/s	Avg BW (bps)	Max BW (bps)	Max Burst	Burst Alarms	Max Buffers (B)
fe80::3cf5:c269:8ac0:c61b	5353	ff02::fb	5353	4	3.85	2835	0	1 / 100ms	0	92
fe80::3cf5:c269:8ac0:c61b	64494	ff02::1:3	5355	1	0.00	0	0	1 / 100ms	0	86
fe80::3cf5:c269:8ac0:c61b	60703	ff02::1:3	5355	1	0.00	0	0	1 / 100ms	0	86
fe80::14ce:86aa:c40:be58	5353	ff02::fb	5353	3	0.75	1039	0	1 / 100ms	0	174
2a00:ca8:a1f:ea5b::1003	5353	ff02::fb	5353	2	1.93	1758	0	1 / 100ms	0	114
192.168.100.254	5353	224.0.0.251	5353	4	3.85	2711	0	1 / 100ms	0	82
192.168.100.237	5353	224.0.0.251	5353	3	0.75	919	0	1 / 100ms	0	154
192.168.100.100	57757	239.255.255.250	1900	1	0.00	0	0	1 / 100ms	0	212
192.168.100.100	5353	224.0.0.251	5353	5	4.81	2819	12 k	2 / 100ms	0	72
192.168.100.100	64494	224.0.0.252	5355	1	0.00	0	0	1 / 100ms	0	66
192.168.100.100	60703	224.0.0.252	5355	1	0.00	0	0	1 / 100ms	0	66
192.168.100.1	36031	224.0.0.251	5353	1	0.00	0	0	1 / 100ms	0	88
192.168.100.1	46340	224.0.0.251	5353	1	0.00	0	0	1 / 100ms	0	88

Obrázek 13: Tabulka multicastových přenosů s jejich statistikami

Detailnější zobrazení těchto obrázků je zobrazeno v příloze **C.1**.

5.6.2 Analýza provozu a zhodnocení implementace ve virtualizovaném prostředí

Popis postupu implementace

1. Instalace služby Avahi na všechny virtuální Ubuntu servery.

```
student@serverVM01:~$ sudo apt-get install avahi-daemon avahi-auotipd
    avahi-dnsconfd avahi-ui-utils avahi-utils -y
student@serverVM02:~$ sudo apt-get install avahi-daemon avahi-auotipd
    avahi-dnsconfd avahi-ui-utils avahi-utils -y
student@serverVM03:~$ sudo apt-get install avahi-daemon avahi-auotipd
    avahi-dnsconfd avahi-ui-utils avahi-utils -y
```

2. Dodatečné nastavení parametrů v Avahi Daemon konfiguračním souboru.

```
    student@serverVM01:/etc$ sudo nano /etc/avahi/avahi-daemon.conf
[server]
host-name=serverVM01
domain-name=local
use-ipv4=no
use-ipv6=yes

    student@serverVM02:/etc$ sudo nano /etc/avahi/avahi-daemon.conf
[server]
host-name=serverVM02
domain-name=local
use-ipv4=no
use-ipv6=yes

    student@serverVM03:/etc$ sudo nano /etc/avahi/avahi-daemon.conf
[server]
host-name=serverVM03
domain-name=local
use-ipv4=no
use-ipv6=yes
```

3. Nastavení sdílených služeb na virtuálním serveru 2.

```
    student@serverVM02:/etc$ sudo nano /etc/avahi/services/_smb.service

<?xml version="1.0" standalone='no'?><!--*-nxml-*-->
<!DOCTYPE service-group SYSTEM "avahi-service.dtd">
<service-group>
  <name replace-wildcards="yes">serverVM02 (%h) over Avahi - SMB,HTTP</
     name>
  <service>
    <type>_ssh._tcp</type>
    <port>22</port>
  </service>

    <service>
    <type>_smb._tcp</type>
        <port>139</port>
        <port>445</port>
  </service>

</service-group>
```

4. Oprava chyby v nsswitch konfiguračním souboru.

```
    student@serverVM01:/etc$ sudo nano /etc/nsswitch.conf
hosts:          files mdns4_minimal mdns6_minimal [NOTFOUND=return] dns

    student@serverVM02:/etc$ sudo nano /etc/nsswitch.conf
hosts:          files mdns4_minimal mdns6_minimal [NOTFOUND=return] dns

    student@serverVM03:/etc$ sudo nano /etc/nsswitch.conf
hosts:          files mdns4_minimal mdns6_minimal [NOTFOUND=return] dns
```

5. Instalace služby SMB a CIFS na virtuálním serveru 2.

```
    student@serverVM02:/etc$ sudo apt-get install samba cifs-utils -y
```

6. Přidání uživatele pro SMB.

```
student@serverVM02:/home $ sudo chown student student
student@serverVM02:/home $ sudo smbpasswd -a student
student@serverVM02:/home $ sudo smbpasswd -e student
student@serverVM02:/home $ sudo chown student:sambashare student
```

7. Přidání konfigurace SMB do smb.conf konfiguračního souboru.

```
student@serverVM02:/home $ sudo nano /etc/samba/smb.conf

[Zeroconf Filesharing]
        path = /home/student
        writeable = no
        browseable = yes
        read only = no
        write list = student
        read list = student
        valid users = student
        security = user
```

8. Povolení síťových pravidel.

```
student@serverVM02:/etc/samba $ sudo ufw allow samba
Rule added
Rule added (v6)

student@serverVM02:/etc/samba $ sudo ufw allow CIFS
Rule added
Rule added (v6)
```

9. Připojení SMB sdílení pomocí CLI na virtuálním serveru 1.

```
student@serverVM01:/etc$ sudo su
root@serverVM01:/mnt# mkdir /mnt/samba
root@serverVM01:/mnt# mount -t cifs -o user=student //serverVM02.local/
    home/student /mnt/samba
```

Ověření funkčnosti

Prohledávání služeb s detailním popisem o službě a stanici (port, hostname a IPv6 adresa). Viz příloha **B.10**

Otestování implementace pomocí pingu vzájemně mezi stanicemi v ".local"doméně.

```
    student@serverVM01:/etc/avahi$ ping serverVM02.local
PING serverVM02.local(2001:718:1001:2c6::117 (2001:718:1001:2c6::117)) 56 data
    bytes
64 bytes from 2001:718:1001:2c6::117 (2001:718:1001:2c6::117): icmp_seq=1 ttl
    =64 time=0.478 ms
64 bytes from 2001:718:1001:2c6::117 (2001:718:1001:2c6::117): icmp_seq=2 ttl
    =64 time=0.414 ms
64 bytes from 2001:718:1001:2c6::117 (2001:718:1001:2c6::117): icmp_seq=3 ttl
    =64 time=0.709 ms
64 bytes from 2001:718:1001:2c6::117 (2001:718:1001:2c6::117): icmp_seq=4 ttl
    =64 time=0.901 ms
--- serverVM02.local ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3026ms
rtt min/avg/max/mdev = 0.414/0.625/0.901/0.194 ms

student@serverVM01:/etc/avahi$ ping serverVM03.local
PING serverVM03.local(2001:718:1001:2c6::114 (2001:718:1001:2c6::114)) 56 data
    bytes
64 bytes from 2001:718:1001:2c6::114 (2001:718:1001:2c6::114): icmp_seq=1 ttl
    =64 time=1.05 ms
64 bytes from 2001:718:1001:2c6::114 (2001:718:1001:2c6::114): icmp_seq=2 ttl
    =64 time=0.676 ms
64 bytes from 2001:718:1001:2c6::114 (2001:718:1001:2c6::114): icmp_seq=3 ttl
    =64 time=0.679 ms
64 bytes from 2001:718:1001:2c6::114 (2001:718:1001:2c6::114): icmp_seq=4 ttl
    =64 time=0.675 ms
--- serverVM03.local ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 0.675/0.771/1.057/0.168 ms
```

Ověření implementace pomocí služby Avahi-Resolve, která překládá doménové jména a IP adresy.

```
student@serverVM01:/etc/avahi$ avahi-resolve --n serverVM03.local
serverVM03.local      2001:718:1001:2c6::114
student@serverVM01:/etc/avahi$ avahi-resolve --n serverVM02.local
serverVM02.local      2001:718:1001:2c6::117
```

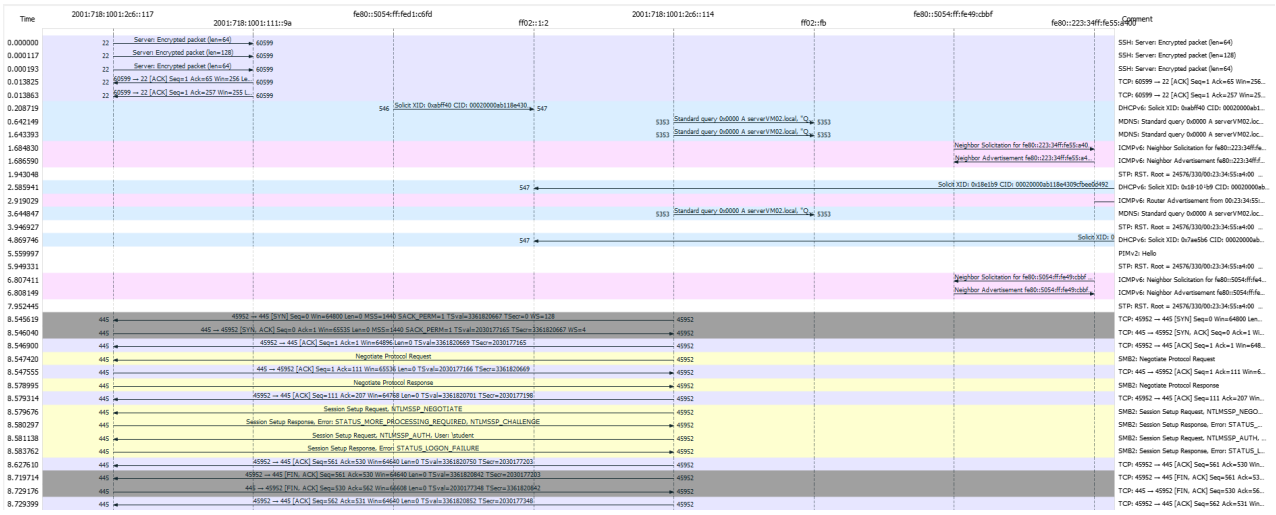
Připojení na SMB sdílenou složku pomocí služby Avahi, výpis z klienta a SMB serveru.

```
root@serverVM01:/mnt# mount -t cifs -o user=student //serverVM02.local/home/
    student /mnt/samba
Password for student@//serverVM02.local/home/student: **************

student@serverVM02:/home$ sudo systemctl status smbd.service
 smbd.service - Samba SMB Daemon
   Loaded: loaded (/lib/systemd/system/smbd.service; enabled; vendor preset:
       enabled)
   Active: active (running) since Sun 2020-05-10 23:17:45 CEST; 1min 51s ago
     Docs: man:smbd(8)
           man:samba(7)
           man:smb.conf(5)
 Main PID: 2269 (smbd)
   Status: "smbd: ready to serve connections..."
    Tasks: 4 (limit: 1108)
   CGroup: /system.slice/smbd.service
           2269 /usr/sbin/smbd --foreground --no-process-group
           2290 /usr/sbin/smbd --foreground --no-process-group
           2291 /usr/sbin/smbd --foreground --no-process-group
           2292 /usr/sbin/smbd --foreground --no-process-group

May 10 23:17:45 serverVM02 systemd[1]: Starting Samba SMB Daemon...
May 10 23:17:45 serverVM02 systemd[1]: Started Samba SMB Daemon.
May 10 23:18:03 serverVM02 smbd[2314]: pam_unix(samba:session): session opened
    for user student by (uid=0)
```

Analýza provozu

Analýza proběhla odchycením provozu pomocí tcpdump a následnou analýzou v programu Wireshark. Na obrázku **14** můžeme vidět úspešně vyhledání služby SMB z GNU/Linux Ubuntu server stanice, připojení pomocí Zeroconf a navázání SMB spojení.

Na druhém obrázku **15** můžeme vidět UDP multicastový přenos, jeho statistiky a rychlosti.



Obrázek 14: Graf zobrazující odchycenou komunikaci při navázání SMB konexe pomocí mDNS

Wireshark · UDP Multicast Streams · virtual.pcap

Source Address	Source Port	Destination Address	Destination Port	Packets	Packets/s	Avg BW (bps)	Max BW (bps)	Max Burst	Burst Alarms	Max Buffers (B)
fe80::5054:ff:fed1:c6fd	546	ff02::1:2	547	1	0.00	0	0	1 / 100ms	0	136
fe80::5054:ff:fe6b:65a9	546	ff02::1:2	547	1	0.00	0	0	1 / 100ms	0	136
fe80::5054:ff:fe4b:818	546	ff02::1:2	547	1	0.00	0	0	1 / 100ms	0	136
2001:718:1001:2c6::114	5353	ff02::fb	5353	3	1.00	767	0	1 / 100ms	0	96

Obrázek 15: Tabulka multicastových přenosů s jejich statistikami

Detailnější zobrazení těchto obrázků je zobrazeno v příloze **C.2**.

5.6.3 Analýza provozu a zhodnocení implementace ve školní laboratoři

Popis postupu implementace

1. Instalace služby Avahi na Raspberry PI a dvou PC v laboratoři.

```
pi@rpi:~$ sudo apt-get install avahi-daemon avahi-auotipd avahi-dnsconfd
    avahi-ui-utils avahi-utils -y
student@pc1:~$ sudo apt-get install avahi-daemon avahi-auotipd avahi-
    dnsconfd avahi-ui-utils avahi-utils -y
student@pc4:~$ sudo apt-get install avahi-daemon avahi-auotipd avahi-
    dnsconfd avahi-ui-utils avahi-utils -y
```

2. Nastavení dodatečných parametrů Avahi Daemona.

```
    pi@rpi:/etc$ sudo nano /etc/avahi/avahi-daemon.conf
[server]
host-name=rpi
domain-name=local
use-ipv4=no
use-ipv6=yes

    student@pc1:/etc$ sudo nano /etc/avahi/avahi-daemon.conf
[server]
host-name=pc1
domain-name=local
use-ipv4=no
use-ipv6=yes

    student@pc4:/etc$ sudo nano /etc/avahi/avahi-daemon.conf
[server]
host-name=pc4
domain-name=local
use-ipv4=no
use-ipv6=yes
```

3. Nastavení sdílených mDNS služeb na Raspberry PI.

```
    pi@rpi:/etc$ sudo nano /etc/avahi/services/_smb.service

<?xml version="1.0" standalone='no'?><!--*-nxml-*-->
<!DOCTYPE service-group SYSTEM "avahi-service.dtd">
<service-group>
  <name replace-wildcards="yes">RPi (%h) over Avahi - SMB,HTTP</name>
  <service>
    <type>_ssh._tcp</type>
    <port>22</port>
  </service>

    <service>
    <type>_smb._tcp</type>
        <port>139</port>
        <port>445</port>
  </service>

</service-group>
```

4. Oprava chyb v nsswitch konfiguraci.

```
    pi@rpi:/etc$ sudo nano /etc/nsswitch.conf
hosts:          files mdns4_minimal mdns6_minimal [NOTFOUND=return] dns

    student@pc1:/etc$ sudo nano /etc/nsswitch.conf
hosts:          files mdns4_minimal mdns6_minimal [NOTFOUND=return] dns

    student@pc4:/etc$ sudo nano /etc/nsswitch.conf
hosts:          files mdns4_minimal mdns6_minimal [NOTFOUND=return] dns
```

5. Instalace SMB služby na Raspberry PI.

```
    pi@rpi:/etc$ sudo apt-get install samba cifs-utils -y
```

6. Nastavení uživatele "pi"pro SMB sdílení.

```
pi@rpi:/home $ sudo chown pi pi
pi@rpi:/home $ sudo smbpasswd -a pi
pi@rpi:/home $ sudo smbpasswd -e pi
pi@rpi:/home $ sudo chown pi:sambashare pi
```

7. Nastavení vlastních parametrů v SMB.conf pro sdílení souborů uživatelem "pi".

```
pi@rpi:/home $ sudo nano /etc/samba/smb.conf

[Zeroconf Filesharing]
        path = /home/pi
        writeable = no
        browseable = yes
        read only = no
        write list = pi
        read list = pi
        valid users = pi
        security = user
```

8. Přidání síťových pravidel pro povolení služeb SMB na Raspberry PI.

```
pi@rpi:/etc/samba $ sudo ufw allow samba
Rule added
Rule added (v6)

pi@rpi:/etc/samba $ sudo ufw allow CIFS
Rule added
Rule added (v6)
```

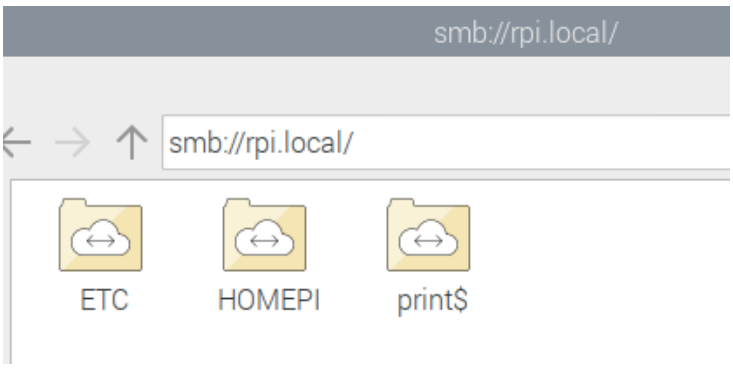
9. Připojení sdílené SMB složky na počítači 1 v laboratoři.

```
student@pc1:/etc$ sudo su
root@pc1:/mnt# mkdir /mnt/samba
root@pc1:/mnt# mount -t cifs -o user=pi //rpi.local/home/pi /mnt/samba
```

Ověření funkčnosti

Připojení na sdílenou složku pomocí služby Avahi a mDNS. Výpis z SMB serveru.

```
pi@rpi:/etc/samba $ sudo systemctl status smbd.service
smbd.service - Samba SMB Daemon
   Loaded: loaded (/lib/systemd/system/smbd.service; enabled; vendor preset:
       enabled)
   Active: active (running) since Mon 2020-02-24 00:23:11 CET; 2 months 16 days
        ago
     Docs: man:smbd(8)
           man:samba(7)
           man:smb.conf(5)
 Main PID: 1945 (smbd)
   Status: "smbd: ready to serve connections..."
    Tasks: 5 (limit: 1599)
   Memory: 16.2M
   CGroup: /system.slice/smbd.service
           1945 /usr/sbin/smbd --foreground --no-process-group
           1947 /usr/sbin/smbd --foreground --no-process-group
           1948 /usr/sbin/smbd --foreground --no-process-group
           1950 /usr/sbin/smbd --foreground --no-process-group
           5910 /usr/sbin/smbd --foreground --no-process-group
Apr 30 12:54:43 rpi smbd[26357]: pam_unix(samba:session): session opened for
    user pi by (uid=0)
Apr 30 15:09:54 rpi smbd[26357]: pam_unix(samba:session): session closed for
    user pi
```



Obrázek 16: Připojení na sdílenou složku na Raspberry PI přes Avahi

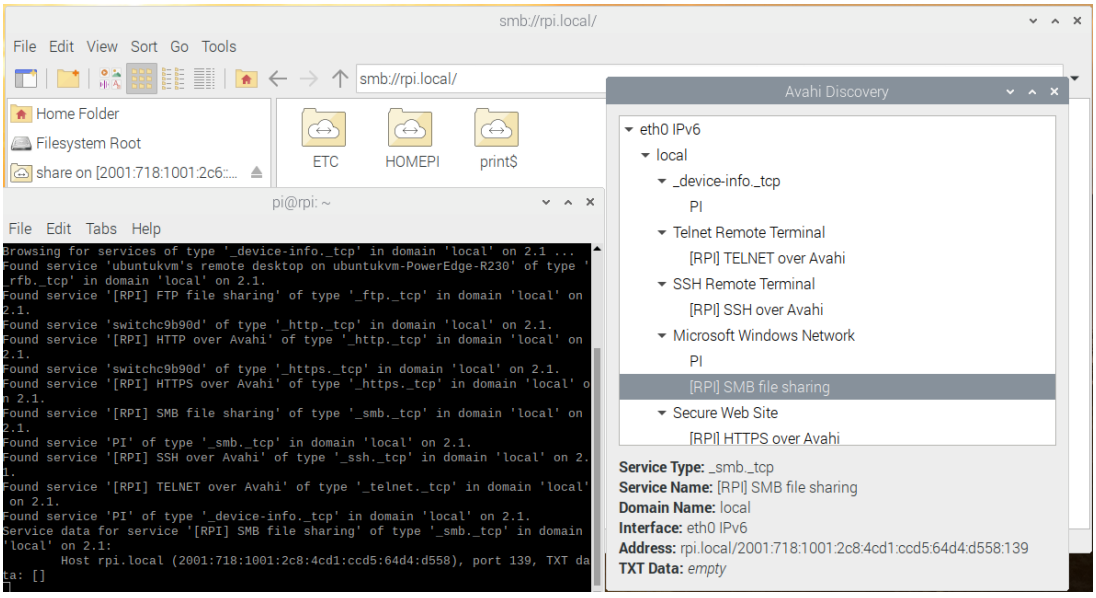
Otestování funkčnosti služby Avahi pingem mezi stanicemi a překladem přes Avahi Resolve.

```
pi@rpi:/etc/samba $ ping pc1.local
PING pc1.local(pc1.local (2001:718:1001:2c8:ffb3:4d09:e5fc:b313%2)) 56 data
    bytes
64 bytes from pc1.local (2001:718:1001:2c8:ffb3:4d09:e5fc:b313): icmp_seq=1 ttl
    =64 time=0.635 ms
64 bytes from pc1.local (2001:718:1001:2c8:ffb3:4d09:e5fc:b313): icmp_seq=2 ttl
    =64 time=0.390 ms
64 bytes from pc1.local (2001:718:1001:2c8:ffb3:4d09:e5fc:b313): icmp_seq=3 ttl
    =64 time=0.387 ms
64 bytes from pc1.local (2001:718:1001:2c8:ffb3:4d09:e5fc:b313): icmp_seq=4 ttl
    =64 time=0.394 ms
64 bytes from pc1.local (2001:718:1001:2c8:ffb3:4d09:e5fc:b313): icmp_seq=5 ttl
    =64 time=0.395 ms
--- pc1.local ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 156ms
rtt min/avg/max/mdev = 0.387/0.440/0.635/0.098 ms

pi@rpi:/etc/samba $ ping pc4.local
PING pc4.local(pc4.local (2001:718:1001:2c8:f83f:65f7:d0c6:6f50%2)) 56 data
    bytes
64 bytes from pc4.local (2001:718:1001:2c8:f83f:65f7:d0c6:6f50): icmp_seq=1 ttl
    =64 time=0.615 ms
64 bytes from pc4.local (2001:718:1001:2c8:f83f:65f7:d0c6:6f50): icmp_seq=2 ttl
    =64 time=0.271 ms
64 bytes from pc4.local (2001:718:1001:2c8:f83f:65f7:d0c6:6f50): icmp_seq=3 ttl
    =64 time=0.372 ms
64 bytes from pc4.local (2001:718:1001:2c8:f83f:65f7:d0c6:6f50): icmp_seq=4 ttl
    =64 time=0.364 ms
64 bytes from pc4.local (2001:718:1001:2c8:f83f:65f7:d0c6:6f50): icmp_seq=5 ttl
    =64 time=0.370 ms
--- pc4.local ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 116ms
rtt min/avg/max/mdev = 0.271/0.398/0.615/0.116 ms
```

Ověření funkčnosti služby Avahi prohledáním služeb přes Avahi-Browse a Avahi-Resolve s detailním výpisem v příloze **B.11**.

```
pi@rpi:/etc/samba $ avahi-resolve --n pc1.local
pc1.local       2001:718:1001:2c8:fc1a:54d9:ad52:150
pi@rpi:/etc/samba $ avahi-resolve --n pc4.local
pc4.local       2001:718:1001:2c8:359a:d002:a630:6060
pi@rpi:/etc/samba $ avahi-resolve --n rpi.local
rpi.local       2001:718:1001:2c8:4cd1:ccd5:64d4:d558

student@pc4:~$ avahi-browse --a
+ enp1s0 IPv6 ubuntukvm's remote desktop on ubuntukvm-PowerEdge-R230 VNC Remote
    Access local
+ enp1s0 IPv6 [RPI] FTP file sharing                FTP File Transfer  local
+ enp1s0 IPv6 switchc9b90d                          Web Site           local
+ enp1s0 IPv6 [RPI] HTTP over Avahi                 Web Site           local
+ enp1s0 IPv6 switchc9b90d                          Secure Web Site    local
+ enp1s0 IPv6 [RPI] HTTPS over Avahi                Secure Web Site    local
+ enp1s0 IPv6 [RPI] SMB file sharing                Microsoft Windows Network
    local
+ enp1s0 IPv6 PI                                   Microsoft Windows Network
    local
+ enp1s0 IPv6 [RPI] SSH over Avahi                  SSH Remote Terminal local
+ enp1s0 IPv6 [RPI] TELNET over Avahi               Telnet Remote Terminal
   local
+ enp1s0 IPv6 PI                                   _device-info._tcp  local
```

Ověření funkčnosti služby Avahi pomocí prohledávání v Avahi-Discover GUI. Viz obrázek **17**



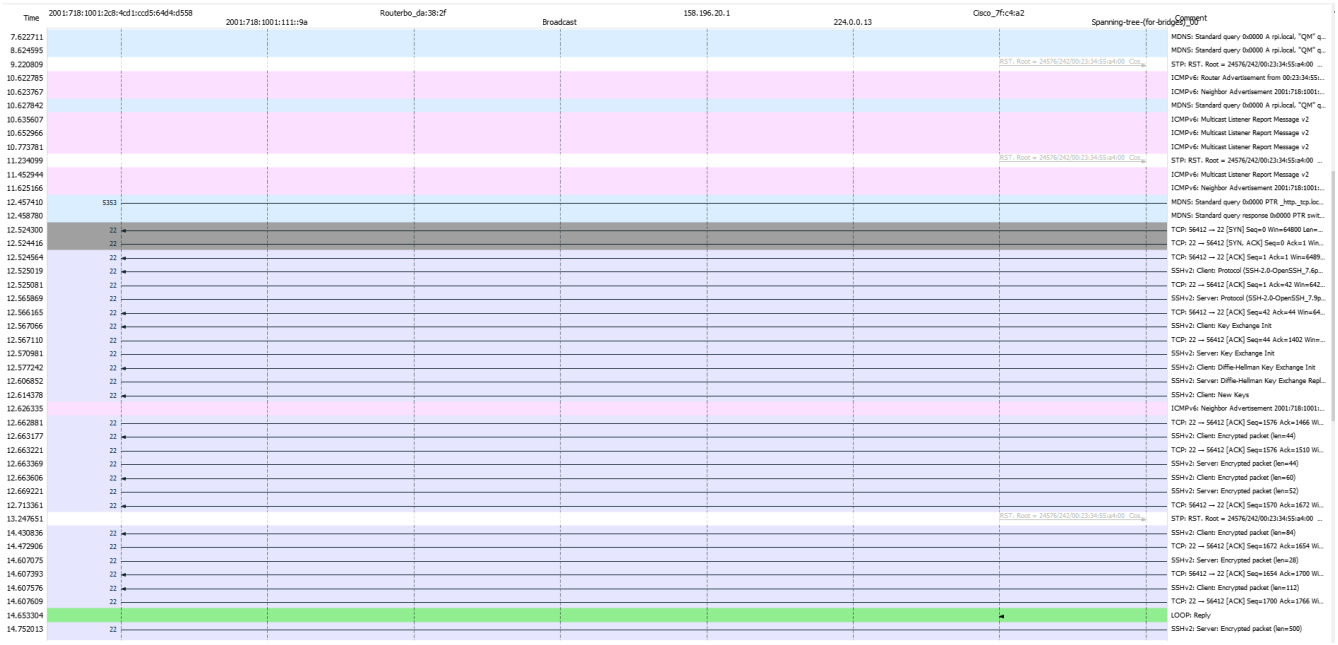
Obrázek 17: Vyhledání služeb pomocí služby Avahi-Discover

Analýza provozu

Analýza proběhla odchycením provozu pomocí tcpdump a následnou analýzou v programu Wireshark. Na obrázku **18** můžeme vidět úspešně vyhledání služby SMB z GNU/Linux Ubuntu stanice, připojení pomocí Zeroconf a navázání SMB spojení.

Na druhém obrázku **19** můžeme vidět UDP multicastový přenos, jeho statistiky a rychlosti.

Detailnější zobrazení těchto obrázků je zobrazeno v příloze **C.3**.



Obrázek 18: Graf zobrazující odchycenou komunikaci při navázání SMB konexe pomocí mDNS

Wireshark · UDP Multicast Streams · embdded.pcap

Source Address	Source Port	Destination Address	Destination Port	Packets	Packets/s	Avg BW (bps)	Max BW (bps)	Max Burst	Burst Alarms	Max Buffers (B)
fe80::d6c9:3cff:fec9:b90d	5353	ff02::fb	5353	2	0.38	850	0	1 / 100ms	0	276
fe80::be67:1cff:fef6:89cd	546	ff02::1:2	547	1	0.00	0	0	1 / 100ms	0	131
2001:718:1001:2c8:ffb3:4d09:e5fc:b313	5353	ff02::fb	5353	3	1.00	710	0	1 / 100ms	0	89
2001:718:1001:2c8:4cd1:ccd5:64d4:d558	5353	ff02::fb	5353	2	0.38	408	0	1 / 100ms	0	132

Obrázek 19: Tabulka multicastových přenosů s jejich statistikami

Závěr

Stěžejní část této bakalářské práce je zaměřena na zprovoznění sdílení souborů pomocí technologie Zeroconf a služeb Samba a CIFS. V první části práce probíhal teoretický rozbor IPv6 adresace, její historie, důvod použití a předpoklady do budoucnosti. V dalších kapitolách probíhal teoretický rozbor technologie Zeroconf, kde byl průběžně rozebrán DNS-SD protokol, mDNS protokol a jako poslední samotný rozbor služeb Avahi a Bonjour. V poslední kapitole byla teoreticky popsána služba Samba a služba CIFS, společně s jejich rozdíly a bezpečností.

V praktické části této bakalářské práce jsme se nejprve zaobírali návrhy implementací pro ověření funkčnosti Avahi, kde byly zvoleny tři typy síťových zapojení. V první implementaci, která byla spíše implementací testovací, byla odzkoušena technologie Zeroconf v prostředí domácí sítě, kde byla využita jak služba Avahi, tak služba Bonjour na operačních systémech Microsoft Windows a GNU/Linux distribuce Raspbian. Druhým typem implementace byla implementace ve školní laboratoři, kde byla odzkoušena technologie Zeroconf v embedded prostředí, pomocí služby Avahi na dvou GNU/Linux operačních systémech, a to operační systém Ubuntu a Raspbian. Třetí typ implementace technologie Zeroconf spočíval v otestování této technologie ve virtualizovaném prostředí na serverových variantách GNU/Linux distribuce Ubuntu. Toto zapojení bylo zrealizováno na virtualizačním serveru poskytnutém univerzitou.

Všechny návrhy implementací obsahují také popis a postup, jak budou jednotlivé služby, nastavení a dodatečná konfigurace probíhat, dle návodů od výrobců těchto služeb a z dalších internetových zdrojů, které se zabývají implementací těchto služeb.

V posledním bodu praktické části je popsán detailně postup jednotlivých implementací i s přiloženými konfiguracemi z jednotlivých zařízení. Dále je ověřena funkčnost těchto implementací a popsána analýza provozu na každé z těchto implementací. Každá implementace obsahuje zhodnocení. Zhodnocení všech implementací dopadlo kladně, technologie Zeroconf, konkrétně služby Avahi a Bonjour, byly bez problému všude nasazeny a jejich chod byl až na menší nedostatky bezproblémový. Bylo potřeba upravit pár dodatečných konfigračních souborů souvisejících s Avahi Daemonem a mDNS NSS knihovnou, ale žádné větší potíže nenastaly. Instalace služeb Samba a CIFS proběhla také bez problému. Všechny GNU/Linux distribuce tuto službu zvládly a její chod byl následně ověřen. Analýza provozu je doplněna o odchycený provoz každé z implementací, kde je možné nalézt grafické zobrazení komunikace a ukázka z náhodného výběrů paketů, které byly odchyceny.

Závěrem mohu konstatovat, že technologie Zeroconf je nedílnou součástí většiny lokálních sítí, které využívají protokoly pro tisk, přenos souborů nebo ovládání chytrých IoT zařízení a většina uživatelů bohužel ani neví, že ji využívají. Služby Avahi a Bonjour začínají být méně

aktualizované a objevují se stále nové chyby v procesech nebo bezpečnostní mezery, které nejsou opraveny. V přílohách této bakalářské práce jsou přiloženy celkové výpisy konfiguračních souborů a velké obrázky analýzy sítí.

Literatura

1. CHESHIRE, Stuart. *DNS Service Discovery (DNS-SD)*. 2020. Dostupné také z: http://www.dns-sd.org/.
2. CHESHIRE, S.; KROCHMAL, M. *rfc6763*. 2013. Technická zpráva. Network Working Group.
3. *DNS Service Discovery and Zero Configuration Networking Overview*. 2020. Dostupné také z: https://macchina.io/docs/00100-DNSSDOverview.html.
4. *Service Discovery*. 2020. Dostupné také z: https://ns1.com/dns-service-discovery.
5. *Introduction to DNS Service Discovery*. 2013. Dostupné také z: https://developer.apple.com/library/archive/documentation/Networking/Conceptual/dns_discovery_api/Introduction.html.
6. *DNS Service Discovery C*. 2020. Dostupné také z: https://developer.apple.com/documentation/dnssd/dns_service_discovery_c.
7. *DNS-based Service Discovery*. 2020. Dostupné také z: https://specs.openstack.org/openstack/api-sig/guidelines/dns-sd.html.
8. Dostupné také z: https://www.researchgate.net/profile/Ala_Al-Fuqaha/publication/279177017/figure/fig3/AS:294412953354 26@1447211683468/Discovering-print-service-by-DNS-SD.png.
9. CHESHIRE, S.; KROCHMAL, M. *rfc6762*. 2013. Technická zpráva. Internet Engineering Task Force (IETF).
10. STEINBERG, Daniel; CHESHIRE, Stuart. *Zero Configuration Networking: The Definitive Guide*. O'Reilly Media, Inc., 2005. ISBN 0596101007.
11. CHESHIRE, Stuart. *Multicast DNS*. 2017. Dostupné také z: http://www.multicastdns.org/.
12. *multicast-dns*. 2020. Dostupné také z: https://www.npmjs.com/package/multicast-dns.
13. *mDNS: What does it do and should I use it*. 2017. Dostupné také z: https://community.ui.com/questions/mDNS-What-does-it-do-and-should-I-use-it/df0bcae0-0f8f-4181-bd15-dfe6b921732b?page=1.
14. ADMIN. *Guide to Multicast DNS (mDNS) security issues*. 2018-07. Dostupné také z: https://kb.iweb.com/hc/en-us/articles/360005117952-Guide-to-Multicast-DNS-mDNS-security-issues.
15. *Multicast DNS and Service Discovery*. 2010. Dostupné také z: https://docs.oracle.com/cd/E19120-01/open.solaris/819-3194/dnsref-28/index.html.
16. *Zeroconf*. 2006-09. Dostupné také z: https://nms.fjfi.cvut.cz/wiki/Zeroconf.

17. PSTREJCZEK. *Multicast DNS i Bonjour/Zeroconf*. 2017-03. Dostupné také z: http://strejczek.com/multicast-dns-i-bonjourzeroconf/.

18. NOVOTNY, Claire. *Zeroconf*. 2020-04. Dostupné také z: https://github.com/novotnyllc/Zeroconf.

19. *Chapter 1. Introduction to Bonjour and Zeroconf*. 2020. Dostupné také z: https://www.oreilly.com/library/view/zero-configuration-networking/0596101007/ch01.html.

20. S. THOMSON, T. Narten; JINMEI, T. *rfc4862*. 2007. Technická zpráva. Network Working Group.

21. MARTASW. *IPv6*. 2020. Dostupné také z: http://martasw.cz/svet_it/ipv6.

22. *Avahi*. 2020-04. Dostupné také z: https://wiki.archlinux.org/index.php/Avahi.

23. *Avahi Notes*. 2019-03. Dostupné také z: http://www.noah.org/wiki/Avahi_Notes.

24. BAYDAN, İsmail. *Linux Avahi Daemon Tutorial With Examples*. 2017-02. Dostupné také z: https://www.poftut.com/linux-avahi-daemon-tutorial-examples/.

25. *Setting up the discovery of a network share / server with Avahi*. 2015-11. Dostupné také z: https://jsherz.com/avahi/service/mdns/cifs/smb/afp/media-server/2015/11/16/avahi-file-share-discovery.html.

26. ŠTRAUCH, Adam. *Avahi: bez konfigurace na síť*. 2020-03. Dostupné také z: https://www.root.cz/clanky/avahi-bez-konfigurace-na-sit/.

27. *Welcome to Avahi*. 2020-02. Dostupné také z: https://www.avahi.org/.

28. KEMP, Juliet. *Using Zeroconf on Linux: What Is It Good For?* 2020. Dostupné také z: http://www.practicallynetworked.com/sharing/configure_and_use_avahi_and_linux.htm.

29. *Bonjour*. 2020. Dostupné také z: https://developer.apple.com/bonjour/.

30. 2020. Dostupné také z: https://developer.apple.com/bonjour/.

31. BURGESS, Phillip. *Bonjour (Zeroconf) Networking for Windows and Linux*. 2020. Dostupné také z: https://learn.adafruit.com/bonjour-zeroconf-networking-for-windows-and-linux.

32. *Zeroconf Kodi Zeroconf XBMC Bonjour*. 2020. Dostupné také z: https://sybu.co.za/wp/xbmc-zeroconf/.

33. *Zeroconf / Bonjour*. 2020. Dostupné také z: https://docs.poppy-project.org/en/installation/install-zeroconf.html.

34. KNOW-HOW. *Bonjour: What's behind the zeroconf implementation*. 2017-08. Dostupné také z: https://www.ionos.co.uk/digitalguide/server/know-how/bonjour-software-for-zero-configuration-networking/.

35. CHESHIRE, Stuart. *How does Zeroconf compare with Viiv/DLNA/DHWG/UPnP?* 2020. Dostupné také z: http://www.zeroconf.org/ZeroconfAndUPnP.html.

36. MARTIN. *ZeroConf vs. UPnP/SSDP: vlastní server*. 2015-04. Dostupné také z: https://forum.root.cz/index.php?topic=11017.0.

37. *Zeroconf vs. UPnP*. 2010-02. Dostupné také z: https://www.wilderssecurity.com/threads/zeroconf-vs-upnp.265610/.

38. LEISTNER, Radovan. *SAMBA*. 2003. Dostupné také z: https://www.fi.muni.cz/~kas/p090/referaty/2003-podzim/skupina10/samba.html.

39. *Síťové protokoly (XVI. část), Protokol SMB*. 2020. Dostupné také z: https://www.banan.cz/serialy/sitove-protokoly/Sitove-protokoly-XVI-cast-Protokol-SMB.

40. TUHÝ, Radan. *NAS: Práce s daty a sdílení pro pokročilé*. 2013-03. Dostupné také z: https://www.svethardware.cz/nas-prace-s-daty-a-sdileni-pro-pokrocile/37490-3.

41. ROBIN, Hack. *Samba*. 2008. Dostupné také z: https://www.svethardware.cz/nas-prace-s-daty-a-sdileni-pro-pokrocile/37490-3.

42. KOZÁK, Ondřej. *Síťové souborové systémy*. 2017. Dostupné také z: https://www.fi.muni.cz/~kas/pv090/referaty/2017-podzim/smb_nfs.html.

43. ROUSE, Margaret. *Server Message Block Protocol (SMB protocol)*. 2007-01. Dostupné také z: https://searchnetworking.techtarget.com/definition/Server-Message-Block-Protocol.

44. *Microsoft SMB Protocol and CIFS Protocol Overview*. 2018-05. Dostupné také z: https://docs.microsoft.com/en-us/windows/win32/fileio/microsoft-smb-protocol-and-cifs-protocol-overview.

45. SOBERS, ROB. *CIFS vs SMB: What's the Difference?* 2020-03. Dostupné také z: https://www.varonis.com/blog/cifs-vs-smb/.

46. *[MS-CIFS]: Common Internet File System (CIFS) Protocol*. 2020-03. Dostupné také z: docs.microsoft.com/en-us/openspecs/windows_protocols/ms-cifs/d416ff7c-c536-406e-a951-4f04b2fd1d2b.

47. *Common Internet File System (CIFS)*. 2012-11. Dostupné také z: https://www.techopedia.com/definition/1867/common-internet-file-system-cifs.

48. DODD, Steve. *Bonjour messages not received if one party has global ipv6 address and one doesn't*. 2019. Dostupné také z: https://bugs.launchpad.net/ubuntu/+source/pidgin/+bug/1841621. Technická zpráva. bugs.launchpad.net.

49. GRIMREAPER11. *Cannot broadcast both on global and link address on same interface*. 2013. Dostupné také z: https://bugs.launchpad.net/ubuntu/+source/avahi/+bug/1102906. Technická zpráva. bugs.launchpad.net.

50. KARN, Phil. *[avahi] IPv6 link local addresses*. 2010. Dostupné také z: https://lists.freedesktop.org/archives/avahi/2010-March/001863.html. Technická zpráva.

51. ZHANG, Nan. *DISCOVERING AND EXPLOITING NOVEL SECURITY VULNERABILITIES IN APPLE ZEROCONF*. 2010-04. Dostupné také z: www.blackhat.com. Technická zpráva. BLACK HAT.

52. SCLAFANI, Pete. *What is IPv6?* 2020-01. Dostupné také z: https://www.6connect.com/resources/what-is-ipv6/.

53. GRAZIANI, R.; SAFARI, an O'Reilly Media Company. *IPv6 Fundamentals: A Straightforward Approach to Understanding IPv6, 2nd Edition*. Cisco Press, 2017. ISBN 9780134670584. Dostupné také z: https://books.google.cz/books?id=Ghx7tAEACAAJ.

54. SATRAPA, Pavel. *IPv6*. 2019. Dostupné také z: https://knihy.nic.cz/files/edice/IPv6-2019.pdf.

55. SATRAPA, Pavel. *Úvod do IPv6*. Dostupné také z: https://www.cesnet.cz/wp-content/uploads/2017/06/Pavel_Satrapa-Uvod_do_IPv6.pdf.

56. S. CHESHIRE, B. Aboba; GUTTMAN, E. *rfc3927*. 2005. Technická zpráva. Network Working Group.

57. THOMSON, S.; HUITEMA, C. *rfc1886*. 1995. Technická zpráva. Network Working Group.

58. SHAW, Keith. *What is IPv6, and why aren't we there yet?* 2018-09. Dostupné také z: https://www.networkworld.com/article/3254575/what-is-ipv6-and-why-aren-t-we-there-yet.html.

59. T. NARTEN, E. Nordmark; SIMPSON, W. *rfc2461*. 1998. Technická zpráva. Network Working Group.

60. *IPv4 Address Report*. 2020-03. Dostupné také z: https://www.potaroo.net/tools/ipv4/.

61. ZDRÁLEK, Jaroslav. *Komunikační sítě I pro integrovanou výuku VUT a VŠB-TUO*. 2016. Dostupné také z: https://lms.vsb.cz/pluginfile.php/1037677/mod_resource/content/6/PKS_IPv6_edice_I_v04.pdf.

62. *Adresy*. 2019-09. Dostupné také z: https://www.ipv6.cz/cs/adresy.

63. HINDEN, R.; DEERING, S. *rfc4291*. 2006. Technická zpráva. Network Working Group.

64. JELÍNEK, Lukáš. *IPV6: CO TO JE A K ČEMU JE TO VŮBEC DOBRÉ?* 2011-02. Dostupné také z: https://jelinek.blog.respekt.cz/ipv6-co-to-je-a-k-cemu-je-to-vubec-dobre/.

65. *RFC 6540 - IPv6 Support Required for All IP-Capable Nodes*. 2012-04. Technická zpráva. Internet Engineering Task Force (IETF).

66. *IPv6*. 2020. Dostupné také z: https://www.facebook.com/ipv6/?tab=ipv6_country.

67. *Google IPv6*. 2020. Dostupné také z: https://www.google.com/intl/en/ipv6/statistics.html#tab=per-country-ipv6-adoption.

68. KAWAMURA, S.; KAWASHIMA, M. *rfc5952*. 2010. Technická zpráva. Network Working Group.

69. *IPv6 - Address Types & Formats*. 2020. Dostupné také z: https://www.tutorialspoint.com/ipv6/ipv6_address_types.htm.

70. *IPv6 Druhy adres (7.díl)*. 2020. Dostupné také z: https://www.banan.cz/serialy/IPv6/IPv6-Druhy-adres-7-dil.

71. ZIMA, Michal. *IPv6 – pokročilé vlastnosti*. 2012. Dostupné také z: https://www.fi.muni.cz/~kas/pv090/referaty/2012-podzim/ct/ipv6.html.

72. *ICMPV6 – TECH DETAILS ADVANTAGES*. 2020. Dostupné také z: https://www.ipv6.com/general/icmpv6-tech-details-advantages/.

A IPv6

Téměř všechna zařízení jsou globálně připojena do internetové komunikace pomocí TCP/IP (Transmission Control Protocol/Internet Protocol). Tento komunikační standart využívá IP adresaci pro specifikaci, identifikaci a lokaci informací a služeb pro všechna zařízení připojena k internetu. Tento standart je využíván již od 80 let minulého století, kdy byl poprvé spuštěn internetový protokol verze 4 (IPv4) a byl zvolen jako primární metoda pro přenos dat mezi zařízeními v síti. Nicméně v 90. letech minulého století již bylo předpovězeno organizací IETF (Internet Engineering Task Force), že adresní prostor tohoto protokolu bude vyčerpán. V reakci na očekávané vyčerpání IPv4 adresace a pokračující expanzi "always-on Internet", organizace IETF vyvinula Internetový Protokol Verze 6 nebo-li IPv6.[52, 53, 54, 55, 20, 56]

Požadavky pro komerční nasazení IPv6:

- rozsáhlejší adresní prostor, který nebude vyčerpán (ideálně),
- více druhů adresace, výběrové(anycast), skupinové(multicast) a individuální(unicast),
- jednotné adresní schéma pro WAN i LAN,
- hierarchické směrování v souladu s hierarchickou adresací,
- zvýšení bezpečnosti (zahrnout do IPv6 mechanismy pro šifrování, autentizaci a sledování cesty k odesilateli),
- podpora pro kritické služby (se zajištěnou kvalitou),
- optimalizace pro směřování (vysokorychlostní),
- automatická konfigurace - plug and play,
- podpora mobility (přenosné počítače, mobily, apod.),
- hladký a plynulý přechod z IPv4 na IPv6. [53, 54, 55, 20, 56]

Požadavky na vývoj nového adresního prostoru nebyly skromné, jeho vývoje se chopili Steven Deering a Robert Hinden, kteří mají hlavní podíl na vytvoření tohoto protokolu. Díky jejich práci vznikla koncem roku 1995 sada RFC definující základ IPv6. Jedná se o RFC s číselným označením 1883 (Internet Protocol, Version 6 Specification).[54, 21, 57, 20, 56]

Bohužel i po definici nového protokolu IPv6 se komerční nasazení nekonalo, protože IPv6 bylo příliž ožehavé téma a chyběly investice. Většina firem se věnovala rozšíření IPv4, kde se začalo používat CIDR adresování, zvýšila se kritéria pro přidělování adres a zavedl se překlad adresace (NAT).[52, 20, 56, 54, 58, 53]

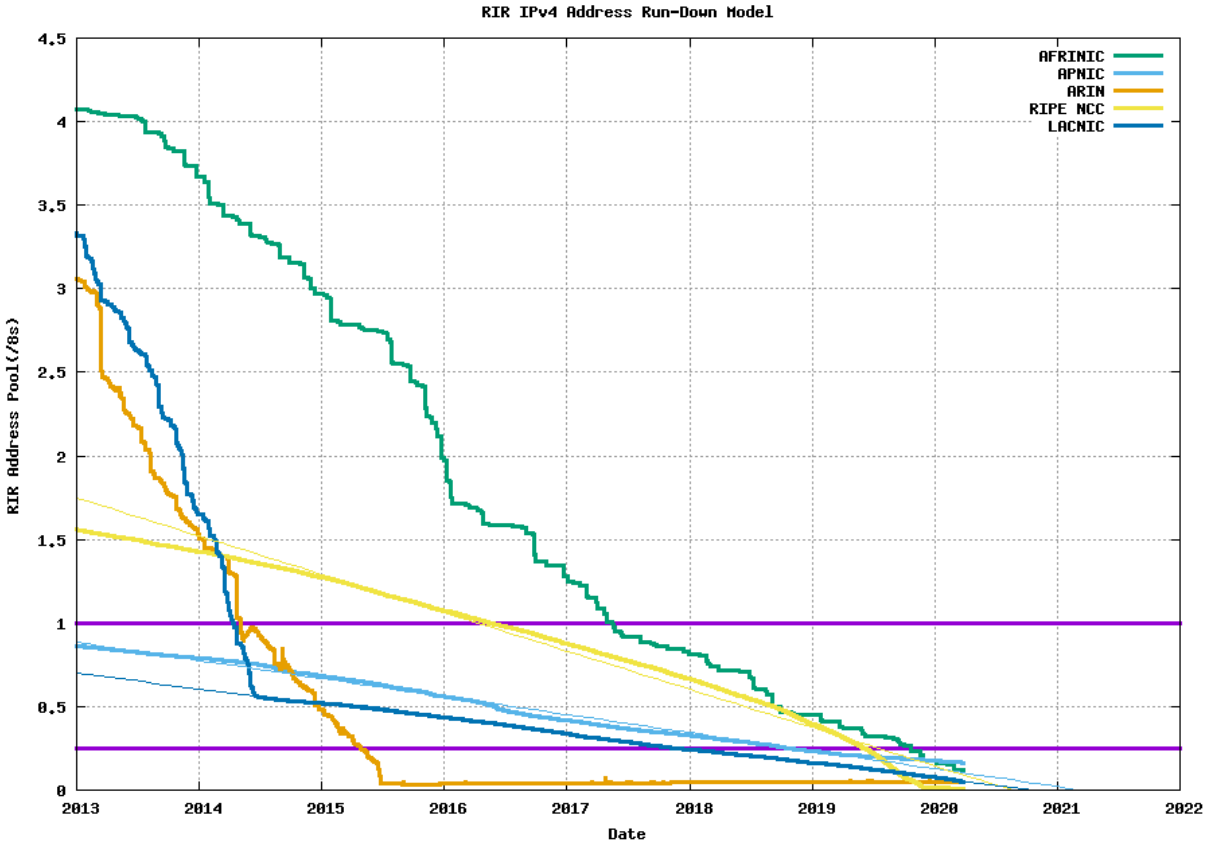
Koncem roku 1998 vyšla revidovaná sada RFC pro IPv6 s číselným označením 2460, která byla postupně doplňována až do komerčního nasazení.[54, 59]

V komerčním nasazení od roku 2006 má tento komunikační protokol nové generace výrazně větší kapacitu adres, než jeho předchůdce, konkrétně IPv6 má k dispozici více než $3.4 * 10^{38}$ unikátních adres, aby vyhověly rostoucí celosvětové poptávce.[52, 55]

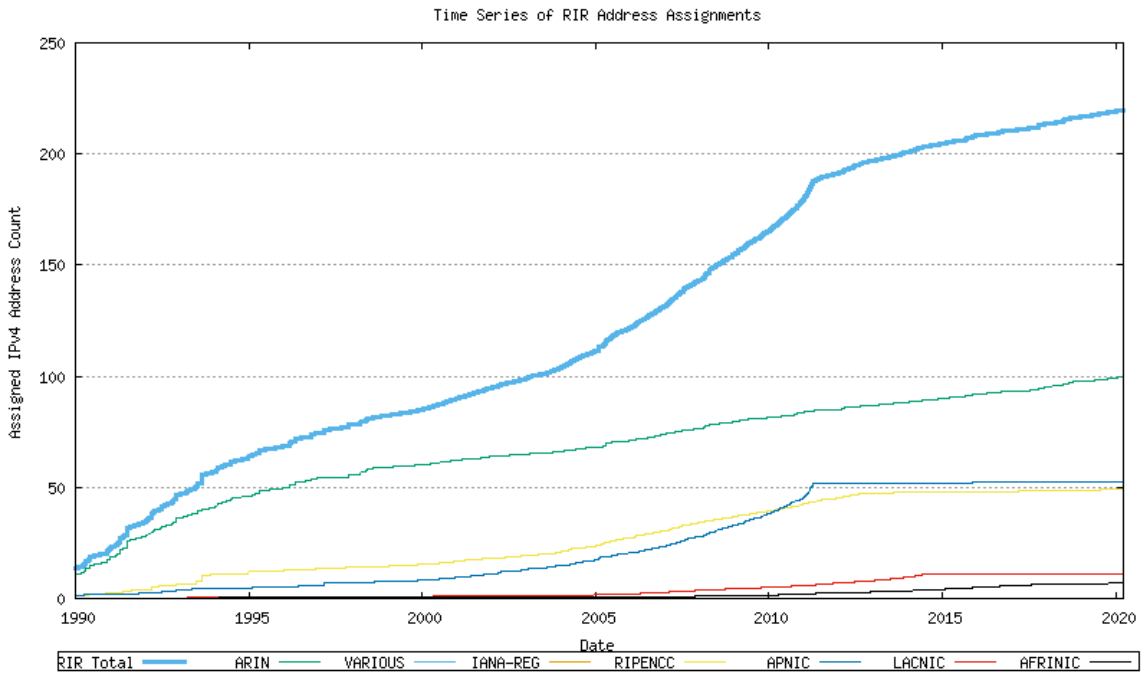
Potřeba protokolu IPv6 je větší, než kdy jindy díky nebývalému růstu internetu a rychlému a neustálému vývoji chytrých telefonů, tabletů, počítačů a dalších online zařízení. V únoru 2011 potvrdila organizace IANA (Internet Assigned Numbers Authority), organizace odpovědná za mezinárodní přidělování IP adres, úplné vyčerpání zdrojů IPv4. Jiné globální organizace také uvedly kritické limity dostupnosti adresy IPv4. V dubnu 2011 se regionální internetový registr v Asii a Tichomoří stal prvním z pěti regionálních registrů IANA, který dosáhl svého limitu adresy IPv4, další následovaly poté. Očekává se, že africký registr pro internetová čísla (AFRINIC), který je posledním nevyčerpaným, bude v blízké budoucnosti následovat.[52, 20, 56, 54]

RIR adresní prostor (data vyčerpání)		
RIR	Datum vyčerpání	Zbývající adresace v RIR poolu (/8s)
APNIC	**19.04.2011**	0.1646
RIPE NCC	**14.08.2012**	0.0101
LACNIC	**10.06.2014**	0.0505
ARIN	**24.08.2015**	0.0002
AFRINIC	**18.06.2020 (předpoklad)**	0.1227

Tabulka 8: Tabulka vyčerpání adresní prostoru RIR [60]



(a) Graf poskytování IP adresace ze zbývajících RIR adresních prostorů.



(b) Graf kumulativního RIR adresního přidělování během let, dle oblastí

Obrázek 20: Grafy přidělování IPv4 adres v RIR adresním prostoru [60]

S blížícím se vyčerpáním zdrojů IPv4 se musí poskytovatelé internetových služeb a podniky na celém světě připravit na zásadní přechod na IPv6. Pochopením rozlišovacích charakteristik IPv6 a souvisejících výzev v oblasti bezpečnosti a nasazení mohou organizace a jejich IT oddělení dohlížet na úspěšnější migrace do tohoto kritického komunikačního protokolu nové generace.[52, 20, 56, 54, 60]

A.1 Hlavní rozdíly mezi IPv4 a IPv6

IPv6 byla navržena tak, aby vyhověla stále rostoucím požadavkům na IP adresaci, která vzhledem ke své binární povaze zůstává omezeným zdrojem. Výsledkem je, že jedním ze základních rozdílů mezi IPv4 a IPv6 je kapacita adres. Nejnovější verze internetového protokolu podporuje více než $3.4 * 10^{38}$ unikátních adres, což představuje výrazné zlepšení kapacity oproti IPv4 o přibližně 4,3 miliardy adres.[52, 61, 62, 63, 64]

Rozšířená kapacita IPv6 adresace je způsobena rozšířenou délkou. IPv4 adresace je 32 bitová, což představuje čtvrtinu délky 128bitových IPv6 adres. Toto pozoruhodné zvýšení umožňuje IPv6 podporovat asi 2 128 jedinečných adres pro každou osobu na planetě. Tato značná kapacita adresace umožní IPv6 splnit rostoucí požadavky na IP adresaci bez obav z vyčerpání zdrojů spojených s IPv4. V síti IPv6 umožní více dostupných adres připojení většího počtu uživatelů a

zařízení k Internetu než kdykoli předtím. [52, 61, 62, 63, 64]

Další zásadní rozdíl mezi IPv4 a IPv6 je založen na technologii a postupy spojené s konfigurací sítě. Protokol Dynamic Host Configuration Protocol, běžně známý jako DHCP, je standardní systém používaný v případě, že zařízení nebo stroj chce automaticky přidělit IP adresaci. V jednom běžném příkladu DHCP používaném v síti, IPv4 DHCP server ve formě routeru spravuje IP informace pro domácí LAN. Počítač, který chce automaticky přidělit IP adresaci, musí zaslat všesměrový požadavek do sítě, kde v našem případě směřovač přiděluje a sleduje informace ohledně DHCP ve své tabulce adres(IP DHCP POOL).[52, 61, 62, 63, 64]

Naproti tomu je IPv6 se svou rozsáhlou kapacitou pro jedinečné přiřazení adresace navržen pro rychlejší a efektivnější přístup k síti. Sítě s IPv6 umožňují notebookům, tabletům a dalším strojům, které hledají přístup k internetu, automatickou konfiguraci pomocí procesu známého jako automatická konfigurace bezstavové adresy (SLAAC). Se SLAAC mohou zařízení, připojená k IPV6, konfigurovat svou vlastní IP adresu a související informace bez podpory ze serveru DHCP.[52, 61, 62, 63, 64]

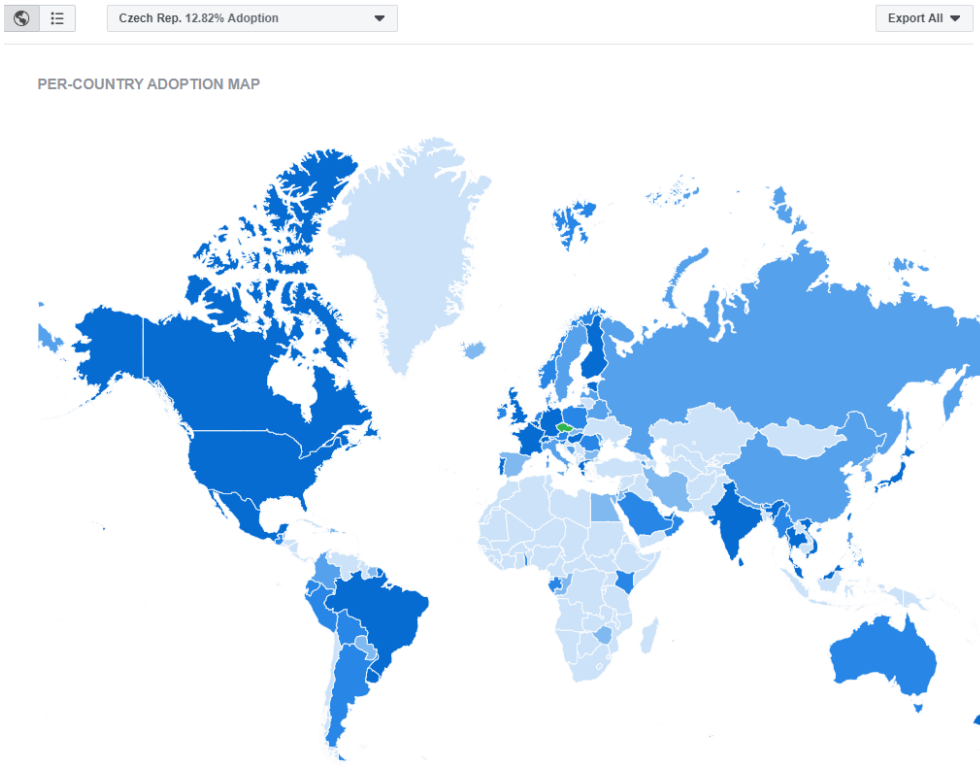
Další změnou u tohoto protokolu je nahrazení všesměrového vysílání, tedy broadcastu, za vícesměrové vysílání (multicast). Na místních linkách lze dosáhnout stejného výsledku pomocí multicastu (skupina all-hosts ff02::1), nicméně většina sítí není připravena pro směřovaní pomocí multicastu, v jednotlivé podsíti bude fungovat, ale globálně nemusí.[52, 61, 62, 63, 64]

A.2 Současný stav

V současné době se organizace IETF snaží o standardizaci protokolu IPv6 a zavedení nových sítí primárně na tomto protokolu, jako budoucnost internetu. V RFC 6540 (IPv6 Support Required for All IP-Capable Nodes) požadují po výrobcích podporu tohoto protokolu.[54, 65]

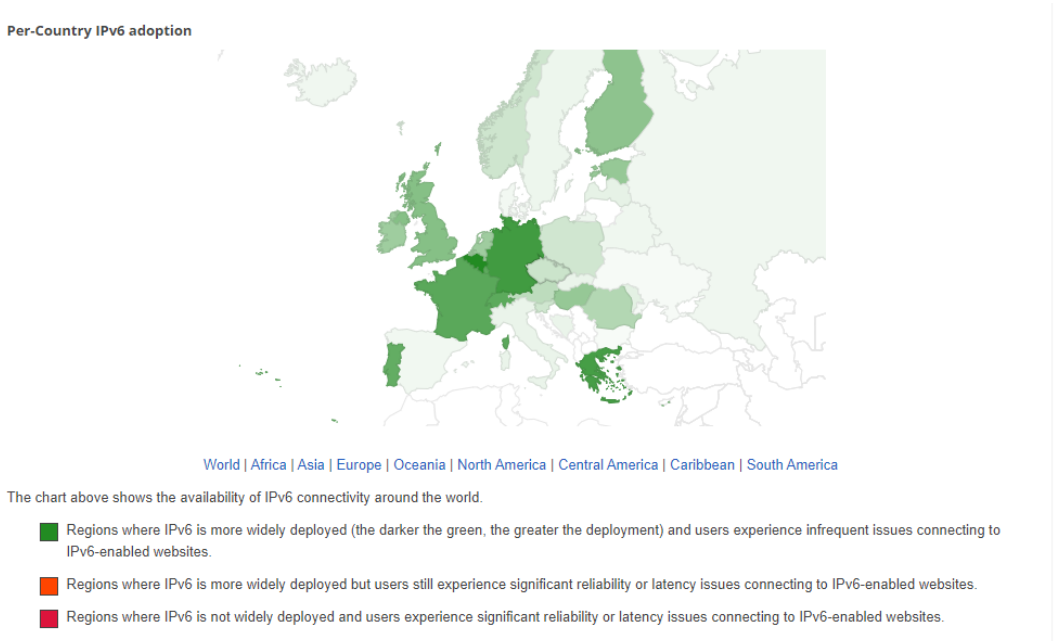
Společnost IAB (Internet Architecture Board), koordinující rozvoj internetu, vydala v roce 2016 prohlášení, ve kterém žádá IETF, aby v nově vytvořených internetových standartech a aktualizacích těch stávajících, přestala předpokládat existenci IPv4. Nové standarty by měly být navrženy tak, aby fungovaly pouze v IPv6 sítích a tento protokol by měl být považován za výchozí.[54]

Současný stav však stále pokulhává za vizí organizace IETF a společnosti IAB. Reálná implementace se pohybuje řádově od 10 až do maximálně 50 procent, nicméně je to slibný začátek a rozhodně se v brzké době dočkáme sítí založených pouze na tomto protokolu.[54]



(a) Mapa procentuálního zobrazení využití IPv6 adres od společnosti Facebook[66]

Obrázek 21: Mapa procentuálního zobrazení využití IPv6 adres od společnosti Facebook[66]



(a) Mapa procentuálního zobrazení využití IPv6 adres[67]

Obrázek 22: Mapa procentuálního zobrazení využití IPv6 adres[67]

A.3 Adresy a Adresní prostor v IPv6

"*Na adresách se nešetřilo. Délka adresy vzrostla na čtyřnásobek původní hodnoty, tedy na 128 bitů. To znamená, že počet všech možných adres se pohybuje v těžko představitelném řádu* 10^{38}. *Na každý milimetr zemského povrchu připadá* $667 * 10^{15}$ *(milionů miliard) adres. Na každého obyvatele planety připadá bezmála 30 tisíc síťových prefixů (každý síťový prefix nabízí 65 tisíc podsítí, z nichž každá umí rozlišit miliardy miliard koncových zařízení). Adres je tedy po všech stránkách dost a měly by vystačit opravdu dlouho.*[62] - ipv6.cz/cs/adresy

Významným rozdílem proti IPv4 je, že zatímco v předchozím protokolu mívalo jedno síťové rozhraní obvykle jen jednu IPv4 adresu, ve světě IPv6 jich mívá celou řadu. Existuje dokonce výčet povinných adres, na nichž počítač či směrovač musí přijímat data."[62] - ipv6.cz/cs/adresy

Základní definice adresace v internetovém protokolu verze 6 je zahrnuta v dokumentu RFC 4291 (IP Version 6 Addressing Architecture), kde můžeme najít informace určující typy adres, jejich délku a podobu. Tento dokument je doplněn o další dokumenty RFC, které důkladně popisují jednotlivé typy adres a dodatečné parametry.[54, 63]

IPv6 adresy jsou 128bitové identifikátory rozhraní a sad rozhraní, existují tři typy adres:

- unicast(individuální): Identifikátor jediného rozhraní. Paket odeslaný na unicast adresu je doručen do identifikovaného rozhraní touto adresou,
- anycast(výběrové): Identifikátor pro sadu rozhraní (obvykle patřící do různých uzlů). Paket odeslaný na anycast adresu je doručen do jednoho z rozhraní identifikován touto adresou ("nejbližší", dle směřovacího protokolu),
- multicast(skupinové): Identifikátor pro sadu rozhraní (obvykle patřící do různých uzlů). Paket odeslaný na multicastovou adresu je doručen do všech rozhraní identifikovaných touto adresou,
- v IPv6 nejsou žádné broadcast(všesměrové) adresy, jejich funkce je nahrazena multicast vysíláním.

[61, 63]

A.3.1 Podoba a zápis adres

Existují tři typy konvenčního zápisu pro reprezentaci IPv6 adres.

1. Preferovaný typ zápisu je X:X:X:X:X:X:X:X, kde písmeno 'X' nahrazuje jedno ze čtyř hexadecimálních čísel. Celkově adresa obsahuje 8 16-bitových částí. Příklad adresy:

   $$ABCD : EF01 : 2345 : 6789 : ABCD : EF01 : 2345 : 6789$$

   $$2001 : DB8 : 0 : 0 : 8 : 800 : 200C : 417A$$

   [54, 63, 61]

2. V zápisu IPv6 adres můžeme nalézt také nuly, které nám umožní zkrátit zápis. Pokud má nějaká skupina na nejvýznamnější pozici nuly, můžeme tyto pozice vynechat ze zápisu. Pokud skupina obsahuje pouze nuly, můžeme tyto pozice zkrátit na jednu nulu. Často také můžeme nalézt více po sobě jdoucích skupin obsahující pouze nuly z důvodu různých metod přidělování IPv6 adres. Tyto řetězce nul můžeme zkrátit pomocí dvou dvojteček '::'. Tento znak v zápisu IPv6 adresy nám indikuje, že jedna nebo více skupin po sobě jdoucích nul byla zkrácena. Tento znak lze použít pouze jednou v zápisu adresy.
   Příklad zkrácení adres:

   - 2001:DB8:0:0:8:800:200C:417A - unicast adresa
   - FF01:0:0:0:0:0:0:101 - multicast adresa
   - 0:0:0:0:0:0:0:1 - loopback adresa
   - 0:0:0:0:0:0:0:0 - nespecifikovaná adresa

   Zkrácená verze:

   - 2001:DB8::8:800:200C:417A - unicast adresa
   - FF01::101 - multicast adresa
   - ::1 - loopback adresa
   - :: - nespecifikovaná adresa

   [54, 63, 61]

3. Alternativní forma, která je někdy výhodnější při práci se smíšeným prostředím IPv4 a IPv6, můžeme zapsat adresu ve formátu X:X:X:X:X:X:d.d.d.d, kde 'X' jsou hexadecimální hodnoty šesti 16-bitových částí adresy vyššího řádu a 'd' jsou desítkové hodnoty čtyř 8-bitových částí adresy nízkého řádu (standardní reprezentace IPv4).
   Příklady zápisu adres:

- 0:0:0:0:0:0:13.1.68.3
- 0:0:0:0:0:FFFF:129.144.52.38

Zkrácená verze:

- ::13.1.68.3
- ::FFFF:129.144.52.38

[54, 63, 61]

V roce 2010 bylo vydáno RFC 5952, které zavedlo kanonický zápis IPv6 adres, který se snaží snížit polymorfii. Byly definovány pravidla, která říkají, že ať už je na vstupu jakýkoliv tvar adresy, musí na jeho výstupu být tvar kanonický.[62, 68]
Jedná se zejména o tyto pravidla:

- hexadecimální znaky/číslice se zapisují pouze malými písmeny,
- pokud má nějaká skupina na nejvýznamnější pozici nuly, musíme tyto pozice vynechat ze zápisu,
- využití zkráceného zápisu pomocí '::' musí mít co největší efekt (využití všech sousedních nulových skupin, musí být použita nejdelší sekvence nul v adrese z levé strany a nesmí obsahovat pouze jednu skupinu).

[61, 62, 63, 68, 69, 70, 71, 72]

A.3.2 Typy adres

Díky velkému adresnímu prostoru, který má IPv6 k dispozici, vzniklo hned několik typů pro sdružování adres se stejnou charakteristikou. Příslušnost k jednotlivým typům adresy určuje prefix. Základní rozdělení je uvedeno v tabulce **9**. Drtivou většinu adresního prostoru zabírají adresy globální individuální adresy, tedy celosvětově jednoznačné, avšak tento prostor je pouze minimálně využit. Většina z tohoto prostoru se ponechává jako rezerva pro budoucí RFC. Zatím se využívá pouze prefix 2000::/3.[61, 62]

Tabulka 9: Základní rozdělení IPv6 prefixů.

prefix	**význam**
::/128	nedefinovaná adresa
::1/128	smyčka (loopback)
fc00::/7	unikátní individuální lokální
fe80::/10	individuální lokální linkové
ff00::/8	skupinové adresy
ostatní	individuální adresy
známé prefixy	
64:ff9b::/96	adresy s vloženým IPv4
64:ff9b:1::/48	lokální adresy pro přechodové mechanismy
2001::/32	Teredo
2001:db8::/32	adresy pro příklady v dokumentech
2002::/16	6to4

B Konfigurace

B.1 Avahi-Daemon

```
# This file is part of avahi.
#
# avahi is free software; you can redistribute it and/or modify it
# under the terms of the GNU Lesser General Public License as
# published by the Free Software Foundation; either version 2 of the
# License, or (at your option) any later version.
#
# avahi is distributed in the hope that it will be useful, but WITHOUT
# ANY WARRANTY; without even the implied warranty of MERCHANTABILITY
# or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public
# License for more details.
#
# You should have received a copy of the GNU Lesser General Public
# License along with avahi; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307
# USA.

# See avahi-daemon.conf(5) for more information on this configuration
# file!

[server]
#host-name=foo
#domain-name=local
#browse-domains=0pointer.de, zeroconf.org
use-ipv4=yes
use-ipv6=yes
#allow-interfaces=eth0
#deny-interfaces=eth1
#check-response-ttl=no
#use-iff-running=no
#enable-dbus=yes
#disallow-other-stacks=no
#allow-point-to-point=no
#cache-entries-max=4096
#clients-max=4096
```

```
#objects-per-client-max=1024
#entries-per-entry-group-max=32
ratelimit-interval-usec=1000000
ratelimit-burst=1000

[wide-area]
enable-wide-area=yes

[publish]
#disable-publishing=no
#disable-user-service-publishing=no
#add-service-cookie=no
#publish-addresses=yes
publish-hinfo=no
publish-workstation=no
#publish-domain=yes
#publish-dns-servers=192.168.50.1, 192.168.50.2
#publish-resolv-conf-dns-servers=yes
#publish-aaaa-on-ipv4=yes
#publish-a-on-ipv6=no

[reflector]
#enable-reflector=no
#reflect-ipv=no

[rlimits]
#rlimit-as=
#rlimit-core=0
#rlimit-data=8388608
#rlimit-fsize=0
#rlimit-nofile=768
#rlimit-stack=8388608
#rlimit-nproc=3
```

Výpis 15: Výpis konfiguračního souboru avahi-daemon.conf

B.2 Avahi-Autoipd

```
#!/bin/sh

# This file is part of avahi.
#
# avahi is free software; you can redistribute it and/or modify it
# under the terms of the GNU Lesser General Public License as
# published by the Free Software Foundation; either version 2 of the
# License, or (at your option) any later version.
#
# avahi is distributed in the hope that it will be useful, but WITHOUT
# ANY WARRANTY; without even the implied warranty of MERCHANTABILITY
# or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public
# License for more details.
#
# You should have received a copy of the GNU Lesser General Public
# License along with avahi; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307
# USA.

set -e

# Command line arguments:
#   $1 event that happened:
#         BIND:    Successfully claimed address
#         CONFLICT: An IP address conflict happened
#         UNBIND:  The IP address is no longer needed
#         STOP:    The daemon is terminating
#   $2 interface name
#   $3 IP adddress

PATH="$PATH:/usr/bin:/usr/sbin:/bin:/sbin"

# Use a different metric for each interface, so that we can set
# identical routes to multiple interfaces.

METRIC=$((1000 + `cat "/sys/class/net/$2/ifindex" 2>/dev/null || echo 0`))
```

```
if [ -x /bin/ip -o -x /sbin/ip ] ; then

    # We have the Linux ip tool from the iproute package

    case "$1" in
        BIND)
            ip addr flush dev "$2" label "$2:avahi"
            ip addr add "$3"/16 brd 169.254.255.255 label "$2:avahi" scope link
                dev "$2"
            ip route add default dev "$2" metric "$METRIC" scope link ||:
            ;;

        CONFLICT|UNBIND|STOP)
            ip route del default dev "$2" metric "$METRIC" scope link ||:
            ip addr del "$3"/16 brd 169.254.255.255 label "$2:avahi" scope link
                dev "$2"
            ;;

        *)
            echo "Unknown event $1" >&2
            exit 1
            ;;
    esac

elif [ -x /bin/ifconfig -o -x /sbin/ifconfig ] ; then

    # We have the old ifconfig tool

    case "$1" in
        BIND)
            ifconfig "$2:avahi" inet "$3" netmask 255.255.0.0 broadcast
                169.254.255.255 up
            route add default dev "$2:avahi" metric "$METRIC" ||:
            ;;

        CONFLICT|STOP|UNBIND)
            route del default dev "$2:avahi" metric "$METRIC" ||:
            ifconfig "$2:avahi" down
            ;;
```

```
        *)
            echo "Unknown event $1" >&2
            exit 1
            ;;
    esac

else

    echo "No network configuration tool found." >&2
    exit 1

fi

exit 0
```

Výpis 16: Výpis programu pro automatické přidělení IPv4 LL adres avahi-autoipd.action

B.3 Avahi-Dnsconfd

```
#!/bin/sh

# This file is part of avahi.
#
# avahi is free software; you can redistribute it and/or modify it
# under the terms of the GNU Lesser General Public License as
# published by the Free Software Foundation; either version 2 of the
# License, or (at your option) any later version.
#
# avahi is distributed in the hope that it will be useful, but WITHOUT
# ANY WARRANTY; without even the implied warranty of MERCHANTABILITY
# or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public
# License for more details.
#
# You should have received a copy of the GNU Lesser General Public
# License along with avahi; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307
# USA.

set -e

test "x$AVAHI_INTERFACE" != "x"

# Command line arguments:
#   $1 "+" if a new DNS server was found, "-" if one was removed
#   $2 DNS Server address
#   $3 interface index where this server was found on
#   $4 protocol number where this server was found on

# Available environment variables:
#
#   $AVAHI_INTERFACE            The interface name where this DNS server was
    found on
#   $AVAHI_INTERFACE_DNS_SERVERS A whitespace seperated list of DNS servers on
    $AVAHI_INTERFACE
#   $AVAHI_DNS_SERVERS          The complete list of all DNS servers found on
    all interfaces
```

```
if [ -x /sbin/netconfig ]; then
    # SUSE method on 11.1+
    if [ -n "$AVAHI_INTERFACE_DNS_SERVERS" ]; then
   /sbin/netconfig modify -s avahi -i "$AVAHI_INTERFACE" <<-EOF
   INTERFACE='$AVAHI_INTERFACE'
   DNSSERVERS='$AVAHI_INTERFACE_DNS_SERVERS'
   EOF
    else
   /sbin/netconfig remove -s avahi -i "$AVAHI_INTERFACE"
    fi
elif [ -x /sbin/modify_resolvconf ] ; then
    # method for SUSE <= 11.0
    if [ -n "$AVAHI_DNS_SERVERS" ]; then
   /sbin/modify_resolvconf modify -s avahi -t - -p avahi-dnsconfd -n "
      $AVAHI_DNS_SERVERS" <<-EOF
   if you don't like avahi to update your Nameservers
   disable the avahi-dnsconfd init script
   EOF
    else
   /sbin/modify_resolvconf restore -s avahi
    fi
elif [ -x /sbin/resolvconf ] ; then

    # We have Debian's resolvconf tool

    if [ "x$AVAHI_INTERFACE_DNS_SERVERS" = "x" ] ; then
        /sbin/resolvconf -d "$AVAHI_INTERFACE.avahi"
    else
        for n in $AVAHI_INTERFACE_DNS_SERVERS ; do
            echo "nameserver $n"
        done | /sbin/resolvconf -a "$AVAHI_INTERFACE.avahi"
    fi
else

    # No resolvconf tool available

    if [ "x$AVAHI_DNS_SERVERS" = "x" ] ; then
```

```
        test -f /etc/resolv.conf.avahi && mv /etc/resolv.conf.avahi /etc/resolv
            .conf
    else
        test -f /etc/resolv.conf.avahi || mv /etc/resolv.conf /etc/resolv.conf.
            avahi

        for n in $AVAHI_DNS_SERVERS ; do
            echo "nameserver $n"
        done > /etc/resolv.conf
    fi
fi
```

Výpis 17: Výpis programu avahi-dnsconfd.action pro automatické přidávání nových Avahi DNS serverů do resolv.conf

B.4 Avahi SMB service

```
<?xml version="1.0" standalone='no'?>
<!DOCTYPE service-group SYSTEM "avahi-service.dtd">
<service-group>
  <name>[RPI] SMB file sharing</name>
  <service>
    <type>_smb._tcp</type>
    <port>139</port>
  </service>
</service-group>
```

Výpis 18: Výpis konfiguračního souboru pro SMB službu v Avahi

B.5 Avahi Hosts

```
127.0.0.1  localhost
::1     localhost ip6-localhost ip6-loopback
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters

127.0.1.1  rpi
```

Výpis 19: Výpis konfiguračního souboru hosts

B.6 mDNS nsswitch

```
# /etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch functionality.
# If you have the 'glibc-doc-reference' and 'info' packages installed, try:
# 'info libc "Name Service Switch"' for information about this file.

passwd:         files
group:          files
shadow:         files
gshadow:        files

hosts:          files mdns4_minimal mdns6 [NOTFOUND=return] mdns4 mdns6 dns
networks:       files

protocols:      db files
services:       db files
ethers:         db files
rpc:            db files

netgroup:       nis
```

Výpis 20: Výpis konfiguračního souboru nsswitch.conf

B.7 DNS resolvconf

```
# Configuration for resolvconf(8)
# See resolvconf.conf(5) for details

resolv_conf=/etc/resolv.conf
# If you run a local name server, you should uncomment the below line and
# configure your subscribers configuration files below.
#name_servers=127.0.0.1

# Mirror the Debian package defaults for the below resolvers
# so that resolvconf integrates seemlessly.
dnsmasq_resolv=/var/run/dnsmasq/resolv.conf
pdnsd_conf=/etc/pdnsd.conf
unbound_conf=/var/cache/unbound/resolvconf_resolvers.conf

nameserver 2001:718:1001::53
nameserver 2606:4700:4700::1001
```

Výpis 21: Výpis konfiguračního souboru resolvconf.conf

B.8 Popis konfigurace SMB.conf

```
;[homes]
;   comment = Home Directories
;   browseable = no
;   read only = yes
;   create mask = 0700
;   directory mask = 0700
;   valid users = %S

;[netlogon]
;   comment = Network Logon Service
;   path = /home/samba/netlogon
;   guest ok = yes
;   read only = yes

;[profiles]
;   comment = Users profiles
;   path = /home/samba/profiles
;   guest ok = no;
;   browseable = no
;   create mask = 0600
;   directory mask = 0700

[printers]
comment = All
Printersbrowseable = no
path = /var/spool/samba
printable = yes
guest ok = no
read only = yes
create mask = 0700

[print$]
comment = Printer Drivers
path = /var/lib/samba/printers
browseable = yes
read only = yes
guest ok = no
```

B.9 Samba SMB.conf

```
#
# Sample configuration file for the Samba suite for Debian GNU/Linux.
#
#
# This is the main Samba configuration file. You should read the
# smb.conf(5) manual page in order to understand the options listed
# here. Samba has a huge number of configurable options most of which
# are not shown in this example
#
# Some options that are often worth tuning have been included as
# commented-out examples in this file.
#  - When such options are commented with ";", the proposed setting
#    differs from the default Samba behaviour
#  - When commented with "#", the proposed setting is the default
#    behaviour of Samba but the option is considered important
#    enough to be mentioned here
#
# NOTE: Whenever you modify this file you should run the command
# "testparm" to check that you have not made any basic syntactic
# errors.

#======================= Global Settings =======================

[global]

## Browsing/Identification ###

# Change this to the workgroup/NT-domain name your Samba server will part of
   workgroup = WORKGROUP

# server string is the equivalent of the NT Description field
        server string = %h server (Samba, Ubuntu)

# Windows Internet Name Serving Support Section:
# WINS Support - Tells the NMBD component of Samba to enable its WINS Server
#   wins support = no
```

```
# WINS Server - Tells the NMBD components of Samba to be a WINS Client
# Note: Samba can be either a WINS Server, or a WINS Client, but NOT both
;   wins server = w.x.y.z

# This will prevent nmbd to search for NetBIOS names through DNS.
   dns proxy = no

#### Networking ####

# The specific set of interfaces / networks to bind to
# This can be either the interface name or an IP address/netmask;
# interface names are normally preferred
;   interfaces = 127.0.0.0/8 eth0

# Only bind to the named interfaces and/or networks; you must use the
# 'interfaces' option above to use this.
# It is recommended that you enable this feature if your Samba machine is
# not protected by a firewall or is a firewall itself. However, this
# option cannot handle dynamic or non-broadcast interfaces correctly.
;   bind interfaces only = yes



#### Debugging/Accounting ####

# This tells Samba to use a separate log file for each machine
# that connects
   log file = /var/log/samba/log.%m

# Cap the size of the individual log files (in KiB).
   max log size = 1000

# If you want Samba to only log through syslog then set the following
# parameter to 'yes'.
#   syslog only = no

# We want Samba to log a minimum amount of information to syslog. Everything
# should go to /var/log/samba/log.{smbd,nmbd} instead. If you want to log
# through syslog you should set the following parameter to something higher.
```

```
   syslog = 0

# Do something sensible when Samba crashes: mail the admin a backtrace
   panic action = /usr/share/samba/panic-action %d


####### Authentication #######

# Server role. Defines in which mode Samba will operate. Possible
# values are "standalone server", "member server", "classic primary
# domain controller", "classic backup domain controller", "active
# directory domain controller".
#
# Most people will want "standalone sever" or "member server".
# Running as "active directory domain controller" will require first
# running "samba-tool domain provision" to wipe databases and create a
# new domain.
   server role = standalone server

# If you are using encrypted passwords, Samba will need to know what
# password database type you are using.
   passdb backend = tdbsam

   obey pam restrictions = yes

# This boolean parameter controls whether Samba attempts to sync the Unix
# password with the SMB password when the encrypted SMB password in the
# passdb is changed.
   unix password sync = yes

# For Unix password sync to work on a Debian GNU/Linux system, the following
# parameters must be set (thanks to Ian Kahan <<kahan@informatik.tu-muenchen.de
    > for
# sending the correct chat script for the passwd program in Debian Sarge).
   passwd program = /usr/bin/passwd %u
   passwd chat = *Enter\snew\s*\spassword:* %n\n *Retype\snew\s*\spassword:* %n
       \n *password\supdated\ssuccessfully* .

# This boolean controls whether PAM will be used for password changes
```

```
# when requested by an SMB client instead of the program listed in
# 'passwd program'. The default is 'no'.
   pam password change = yes

# This option controls how unsuccessful authentication attempts are mapped
# to anonymous connections
   map to guest = bad user

########## Domains ############

#
# The following settings only takes effect if 'server role = primary
# classic domain controller', 'server role = backup domain controller'
# or 'domain logons' is set
#

# It specifies the location of the user's
# profile directory from the client point of view) The following
# required a [profiles] share to be setup on the samba server (see
# below)
;   logon path = \\%N\profiles\%U
# Another common choice is storing the profile in the user's home directory
# (this is Samba's default)
#   logon path = \\%N\%U\profile

# The following setting only takes effect if 'domain logons' is set
# It specifies the location of a user's home directory (from the client
# point of view)
;   logon drive = H:
#   logon home = \\%N\%U

# The following setting only takes effect if 'domain logons' is set
# It specifies the script to run during logon. The script must be stored
# in the [netlogon] share
# NOTE: Must be store in 'DOS' file format convention
;   logon script = logon.cmd

# This allows Unix users to be created on the domain controller via the SAMR
# RPC pipe. The example command creates a user account with a disabled Unix
```

```
# password; please adapt to your needs
; add user script = /usr/sbin/adduser --quiet --disabled-password --gecos "" %u

# This allows machine accounts to be created on the domain controller via the
# SAMR RPC pipe.
# The following assumes a "machines" group exists on the system
; add machine script = /usr/sbin/useradd -g machines -c "%u machine account" -d
    /var/lib/samba -s /bin/false %u

# This allows Unix groups to be created on the domain controller via the SAMR
# RPC pipe.
; add group script = /usr/sbin/addgroup --force-badname %g

############ Misc ############

# Using the following line enables you to customise your configuration
# on a per machine basis. The %m gets replaced with the netbios name
# of the machine that is connecting
;   include = /home/samba/etc/smb.conf.%m

# Some defaults for winbind (make sure you're not using the ranges
# for something else.)
;   idmap uid = 10000-20000
;   idmap gid = 10000-20000
;   template shell = /bin/bash

# Setup usershare options to enable non-root users to share folders
# with the net usershare command.

# Maximum number of usershare. 0 (default) means that usershare is disabled.
;   usershare max shares = 100

# Allow users who've been granted usershare privileges to create
# public shares, not just authenticated ones
   usershare allow guests = yes

#======================= Share Definitions =======================

# Un-comment the following (and tweak the other settings below to suit)
```

```
# to enable the default home directory shares. This will share each
# user's home directory as \\server\username
;[homes]
;   comment = Home Directories
;   browseable = no

# By default, the home directories are exported read-only. Change the
# next parameter to 'no' if you want to be able to write to them.
;   read only = yes

# File creation mask is set to 0700 for security reasons. If you want to
# create files with group=rw permissions, set next parameter to 0775.
;   create mask = 0700

# Directory creation mask is set to 0700 for security reasons. If you want to
# create dirs. with group=rw permissions, set next parameter to 0775.
;   directory mask = 0700

# By default, \\server\username shares can be connected to by anyone
# with access to the samba server.
# Un-comment the following parameter to make sure that only "username"
# can connect to \\server\username
# This might need tweaking when using external authentication schemes
;   valid users = %S

# Un-comment the following and create the netlogon directory for Domain Logons
# (you need to configure Samba to act as a domain controller too.)
;[netlogon]
;   comment = Network Logon Service
;   path = /home/samba/netlogon
;   guest ok = yes
;   read only = yes

# Un-comment the following and create the profiles directory to store
# users profiles (see the "logon path" option above)
# (you need to configure Samba to act as a domain controller too.)
# The path below should be writable by all users so that their
# profile directory may be created the first time they log on
;[profiles]
```

```
;   comment = Users profiles
;   path = /home/samba/profiles
;   guest ok = no
;   browseable = no
;   create mask = 0600
;   directory mask = 0700

[printers]
   comment = All Printers
   browseable = no
   path = /var/spool/samba
   printable = yes
   guest ok = no
   read only = yes
   create mask = 0700

# Windows clients look for this share name as a source of downloadable
# printer drivers
[print$]
   comment = Printer Drivers
   path = /var/lib/samba/printers
   browseable = yes
   read only = yes
   guest ok = no
# Uncomment to allow remote administration of Windows print drivers.
# You may need to replace 'lpadmin' with the name of the group your
# admin users are members of.
# Please note that you also need to set appropriate Unix permissions
# to the drivers directory for these users to have write rights in it
;   write list = root, @lpadmin
```

Výpis 22: Výpis konfiguračního souboru smb.conf

B.10 Ověření funkčnosti virtualizační implementace - Avahi-Browse

```
   root@serverVM01:/mnt# avahi-browse --a --r
+   ens3 IPv6 serverVM03 [52:54:00:84:75:46]          Workstation        local
+   ens3 IPv6 FTP file sharing                        FTP File Transfer  local
+   ens3 IPv6 serverVM02 (serverVM02) over Avahi - SMB,HTTP SSH Remote Terminal
     local
+   ens3 IPv6 serverVM01 (serverVM01) over Avahi - SMB,HTTP SSH Remote Terminal
     local
+   ens3 IPv6 serverVM02 (serverVM02) over Avahi - SMB,HTTP Web Site     local
+   ens3 IPv6 serverVM01 (serverVM01) over Avahi - SMB,HTTP Web Site     local
+   ens3 IPv6 SERVERVM02                              Microsoft Windows Network
     local
+   ens3 IPv6 SERVER                                  Microsoft Windows Network
     local
+   ens3 IPv6 serverVM01 (serverVM01) over Avahi - SMB,HTTP Microsoft Windows
    Network local
=   ens3 IPv6 FTP file sharing                        FTP File Transfer  local
   hostname = [serverVM01.local]
   address = [2001:718:1001:2c6::305]
   port = [21]
   txt = []
=   ens3 IPv6 serverVM01 (serverVM01) over Avahi - SMB,HTTP Microsoft Windows
    Network local
   hostname = [serverVM01.local]
   address = [2001:718:1001:2c6::305]
   port = [445]
   txt = []
=   ens3 IPv6 serverVM01 (serverVM01) over Avahi - SMB,HTTP Web Site     local
   hostname = [serverVM01.local]
   address = [2001:718:1001:2c6::305]
   port = [80]
   txt = []
=   ens3 IPv6 serverVM01 (serverVM01) over Avahi - SMB,HTTP SSH Remote Terminal
     local
   hostname = [serverVM01.local]
   address = [2001:718:1001:2c6::305]
   port = [22]
   txt = []
```

```
=   ens3 IPv6 SERVERVM02                                Microsoft Windows Network
    local
   hostname = [serverVM02.local]
   address = [2001:718:1001:2c6::117]
   port = [445]
   txt = []
=   ens3 IPv6 serverVM02 (serverVM02) over Avahi - SMB,HTTP Web Site     local
   hostname = [serverVM02.local]
   address = [2001:718:1001:2c6::117]
   port = [80]
   txt = []
=   ens3 IPv6 serverVM02 (serverVM02) over Avahi - SMB,HTTP SSH Remote Terminal
    local
   hostname = [serverVM02.local]
   address = [2001:718:1001:2c6::117]
   port = [22]
   txt = []
=   ens3 IPv6 SERVER                                    Microsoft Windows Network
    local
   hostname = [serverVM03.local]
   address = [2001:718:1001:2c6::114]
   port = [445]
   txt = []
=   ens3 IPv6 serverVM03 [52:54:00:84:75:46]            Workstation         local
   hostname = [serverVM03.local]
   address = [2001:718:1001:2c6::114]
   port = [9]
   txt = []
```

B.11 Ověření funkčnosti implementace v laboratoři - Avahi-Browse

```
student@pc4:~$ avahi-browse --a --r
+ enp1s0 IPv6 ubuntukvm's remote desktop on ubuntukvm-PowerEdge-R230 VNC Remote
    Access local
+ enp1s0 IPv6 [RPI] FTP file sharing                  FTP File Transfer  local
+ enp1s0 IPv6 switchc9b90d                            Web Site           local
+ enp1s0 IPv6 [RPI] HTTP over Avahi                   Web Site           local
+ enp1s0 IPv6 switchc9b90d                            Secure Web Site    local
+ enp1s0 IPv6 [RPI] HTTPS over Avahi                  Secure Web Site    local
+ enp1s0 IPv6 [RPI] SMB file sharing                  Microsoft Windows Network
    local
+ enp1s0 IPv6 PI                                      Microsoft Windows Network
    local
+ enp1s0 IPv6 [RPI] SSH over Avahi                    SSH Remote Terminal local
+ enp1s0 IPv6 [RPI] TELNET over Avahi                 Telnet Remote Terminal
   local
+ enp1s0 IPv6 PI                                      _device-info._tcp  local
= enp1s0 IPv6 switchc9b90d                            Web Site           local
   hostname = [switchc9b90d.local]
   address = [192.168.1.254]
   port = [80]
   txt = ["path=/config/authentication_page.htm"]
= enp1s0 IPv6 switchc9b90d                            Secure Web Site    local
   hostname = [switchc9b90d.local]
   address = [192.168.1.254]
   port = [443]
   txt = ["path=/config/authentication_page.htm"]
= enp1s0 IPv6 [RPI] FTP file sharing                  FTP File Transfer  local
   hostname = [rpi.local]
   address = [2001:718:1001:2c8:4cd1:ccd5:64d4:d558]
   port = [21]
   txt = []
= enp1s0 IPv6 PI                                      _device-info._tcp  local
   hostname = [rpi.local]
   address = [2001:718:1001:2c8:4cd1:ccd5:64d4:d558]
   port = [0]
   txt = ["model=MacSamba"]
```

```
= enp1s0 IPv6 [RPI] TELNET over Avahi                    Telnet Remote Terminal
    local
   hostname = [rpi.local]
   address = [2001:718:1001:2c8:4cd1:ccd5:64d4:d558]
   port = [23]
   txt = []
= enp1s0 IPv6 [RPI] SSH over Avahi                       SSH Remote Terminal local
   hostname = [rpi.local]
   address = [2001:718:1001:2c8:4cd1:ccd5:64d4:d558]
   port = [22]
   txt = []
= enp1s0 IPv6 PI                                         Microsoft Windows Network
     local
   hostname = [rpi.local]
   address = [2001:718:1001:2c8:4cd1:ccd5:64d4:d558]
   port = [445]
   txt = []
= enp1s0 IPv6 [RPI] SMB file sharing                     Microsoft Windows Network
     local
   hostname = [rpi.local]
   address = [2001:718:1001:2c8:4cd1:ccd5:64d4:d558]
   port = [139]
   txt = []
= enp1s0 IPv6 [RPI] HTTPS over Avahi                     Secure Web Site    local
   hostname = [rpi.local]
   address = [2001:718:1001:2c8:4cd1:ccd5:64d4:d558]
   port = [443]
   txt = []
= enp1s0 IPv6 [RPI] HTTP over Avahi                      Web Site           local
   hostname = [rpi.local]
   address = [2001:718:1001:2c8:4cd1:ccd5:64d4:d558]
   port = [80]
   txt = []
= enp1s0 IPv6 ubuntukvm's remote desktop on ubuntukvm-PowerEdge-R230 VNC Remote
     Access local
   hostname = [ubuntukvm-PowerEdge-R230.local]
   address = [2001:718:1001:2c8:7d1e:1bac:5038:4677]
   port = [5900]
   txt = []
```

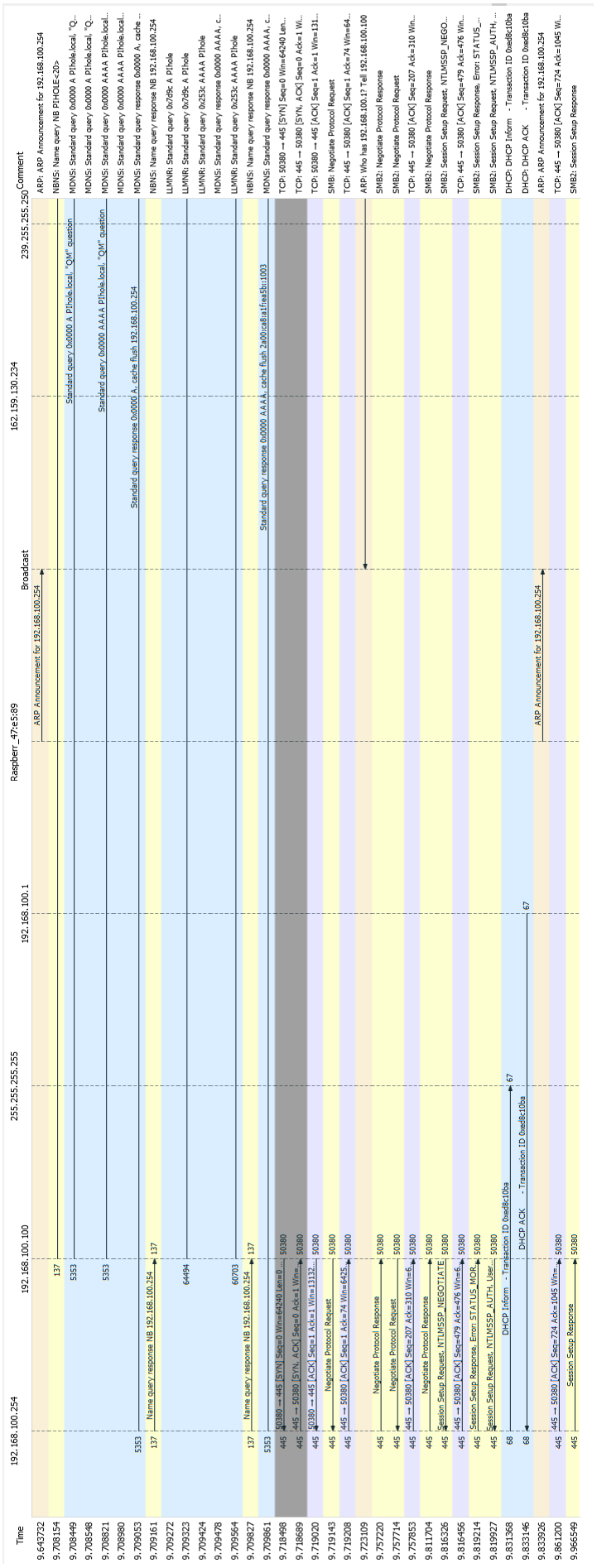
C Obrázky

C.1 Odchycená komunikace v domácí implementaci

Wireshark · UDP Multicast Streams · homelab.pcap

Source Address	Source Port	Destination Address	Destination Port	Packets	Packets/s	Avg BW (bps)	Max BW (bps)	Max Burst	Burst Alarms	Max Buffers (B)
fe80::3cf5:c269:8ac0:c61b	5353	ff02::fb	5353	4	3.85	2835	0	1 / 100ms	0	92
fe80::3cf5:c269:8ac0:c61b	64494	ff02::1:3	5355	1	0.00	0	0	1 / 100ms	0	86
fe80::3cf5:c269:8ac0:c61b	60703	ff02::1:3	5355	1	0.00	0	0	1 / 100ms	0	86
fe80::14ce:86aa:c40:be58	5353	ff02::fb	5353	3	0.75	1039	0	1 / 100ms	0	174
2a00:ca8:a1f:ea5b::1003	5353	ff02::fb	5353	2	1.93	1758	0	1 / 100ms	0	114
192.168.100.254	5353	224.0.0.251	5353	4	3.85	2711	0	1 / 100ms	0	82
192.168.100.237	5353	224.0.0.251	5353	3	0.75	919	0	1 / 100ms	0	154
192.168.100.100	57757	239.255.255.250	1900	1	0.00	0	0	1 / 100ms	0	212
192.168.100.100	5353	224.0.0.251	5353	5	4.81	2819	12 k	2 / 100ms	0	72
192.168.100.100	64494	224.0.0.252	5355	1	0.00	0	0	1 / 100ms	0	66
192.168.100.100	60703	224.0.0.252	5355	1	0.00	0	0	1 / 100ms	0	66
192.168.100.1	36031	224.0.0.251	5353	1	0.00	0	0	1 / 100ms	0	88
192.168.100.1	46340	224.0.0.251	5353	1	0.00	0	0	1 / 100ms	0	88

Obrázek 23: Tabulka multicastových přenosů s jejich statistikami



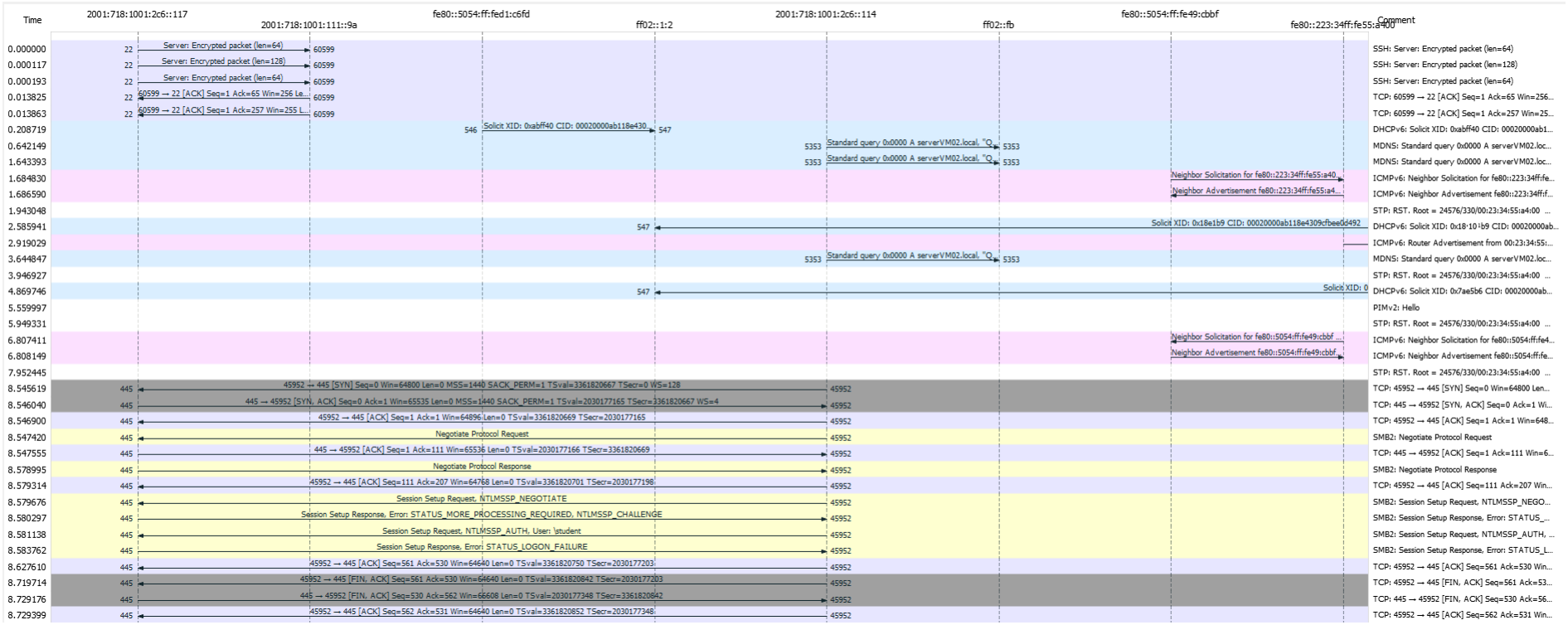
Obrázek 24: Graf zobrazující odchycenou komunikaci při navázání SMB konexe pomocí mDNS

C.2 Odchycená komunikace ve virtuální implementaci

Wireshark · UDP Multicast Streams · virtual.pcap

Source Address	Source Port	Destination Address	Destination Port	Packets	Packets/s	Avg BW (bps)	Max BW (bps)	Max Burst	Burst Alarms	Max Buffers (B)
fe80::5054:ff:fed1:c6fd	546	ff02::1:2	547	1	0.00	0	0	1 / 100ms	0	136
fe80::5054:ff:fe6b:65a9	546	ff02::1:2	547	1	0.00	0	0	1 / 100ms	0	136
fe80::5054:ff:fe4b:818	546	ff02::1:2	547	1	0.00	0	0	1 / 100ms	0	136
2001:718:1001:2c6::114	5353	ff02::fb	5353	3	1.00	767	0	1 / 100ms	0	96

Obrázek 25: Tabulka multicastových přenosů s jejich statistikami



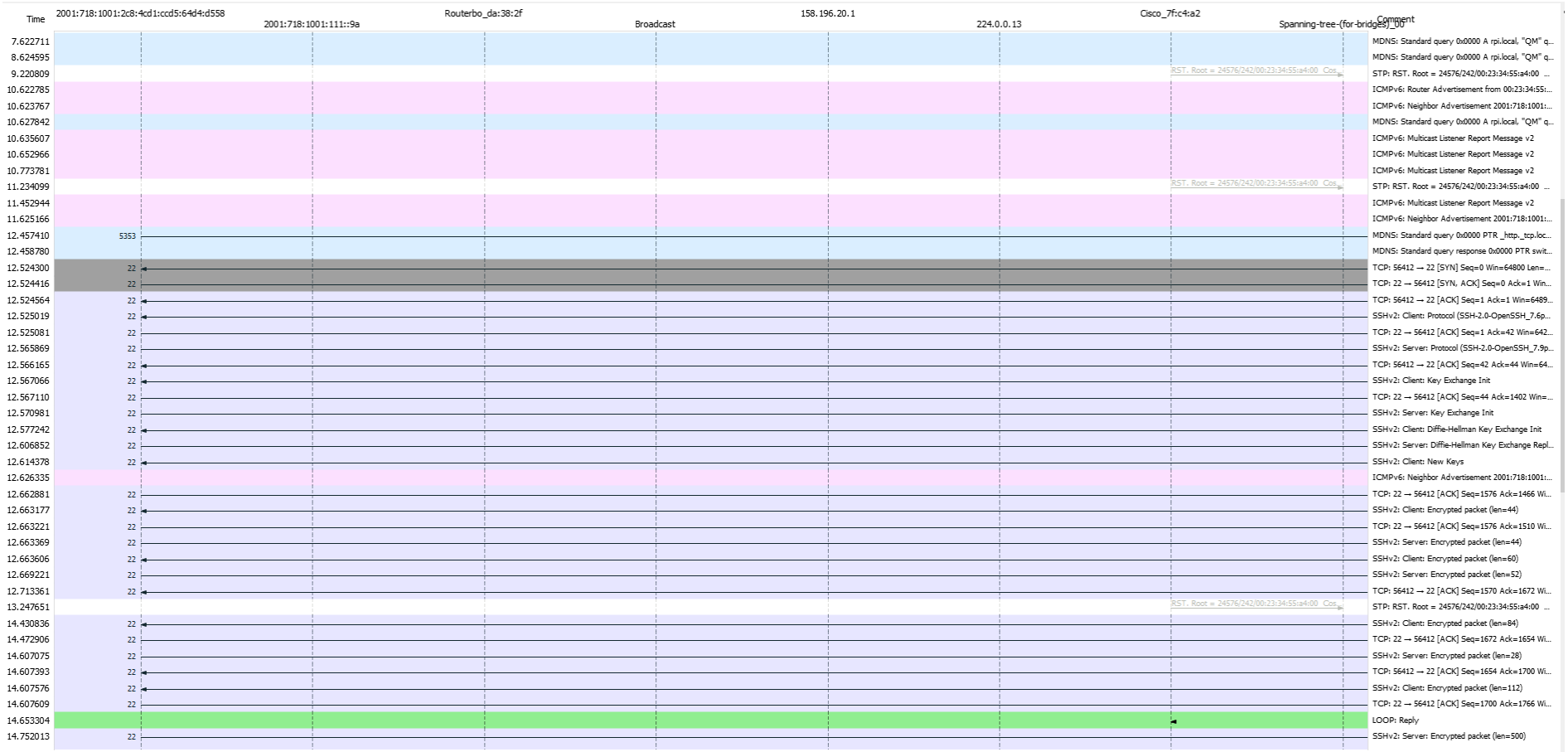
Obrázek 26: Graf zobrazující odchycenou komunikaci při navázání SMB konexe pomocí mDNS

C.3 Odchycená komunikace ve školní laboratoři

Wireshark · UDP Multicast Streams · embdded.pcap

Source Address	Source Port	Destination Address	Destination Port	Packets	Packets/s	Avg BW (bps)	Max BW (bps)	Max Burst	Burst Alarms	Max Buffers (B)
fe80::d6c9:3cff:fec9:b90d	5353	ff02::fb	5353	2	0.38	850	0	1 / 100ms	0	276
fe80::be67:1cff:fef6:89cd	546	ff02::1:2	547	1	0.00	0	0	1 / 100ms	0	131
2001:718:1001:2c8:ffb3:4d09:e5fc:b313	5353	ff02::fb	5353	3	1.00	710	0	1 / 100ms	0	89
2001:718:1001:2c8:4cd1:ccd5:64d4:d558	5353	ff02::fb	5353	2	0.38	408	0	1 / 100ms	0	132

Obrázek 27: Tabulka multicastových přenosů s jejich statistikami



Obrázek 28: Graf zobrazující odchycenou komunikaci při navázání SMB konexe pomocí mDNS